Enforcing topological constraints in random field image segmentation

Chao Chen, Daniel Freedman and Christoph H. Lampert

IST Austria (Institute of Science and Technology Austria)

Am Campus 1

A-3400 Klosterneuburg

Technical Report No. IST-2011-0002

http://pub.ist.ac.at/Pubs/TechRpts/2011/IST-2011-0002.pdf

March 28, 2011

Enforcing topological constraints in random field image segmentation

Chao Chen
IST Austria (Institute of Science and Technology Austria) &
PRIP, Vienna University of Technology, Austria
http://www.ist.ac.at/∼cchen/

Daniel Freedman
School of Engineering,
Bar Ilan University, Ramat Gan, Israel
http://www.eng.biu.ac.il/∼freedmd/

Christoph H. Lampert
IST Austria (Institute of Science and Technology Austria)
http://www.ist.ac.at/∼chl/

Abstract

We introduce TopoCut: a new way to integrate knowledge about topological properties (TPs) into random field image segmentation model. Instead of including TPs as additional constraints during minimization of the energy function, we devise an efficient algorithm for modifying the unary potentials such that the resulting segmentation is guaranteed with the desired properties. Our method is more flexible in the sense that it handles more topology constraints than previous methods, which were only able to enforce pairwise or global connectivity. In particular, our method is very fast, making it for the first time possible to enforce global topological properties in practical image segmentation tasks.

1. Introduction

The topic of this work is the integration of *topological prior knowledge* into *random field* image models. Several image segmentation and restoration tasks can benefit from such a step, *e.g. figure-ground segmentation*, where it is often safe to assume that the foreground region will form a connected component, or medical image segmentation, where topological knowledge about objects, such as hearts, vessels and cortical surfaces, is available.

Discrete random field image models, namely *Markov random fields (MRFs)* [14] and *conditional random fields (CRFs)* [18], are currently amongst the most successful techniques for image restoration and segmentation tasks. Since in this work we only study the task of prediction, not of parameter learning, we will not distinguish between both classes and refer to them jointly as *random field* models. Random field models are popular because they allow simple probabilistic modeling of image properties while at the same time providing very efficient inference tools, in particular the *GraphCut* algorithm [9]. However, in their most common (and most efficiently solvable) *pairwise* form, random field models only encode local interactions between pairs of pixels. This makes it hard for them to express higher-level prior information, such as region shape or higher-order image statistics.

To address this shortcoming, recent work has started to study the integration of *higher-order potentials* into random field models. For reasons of computational tractability, successful approaches in this area either rely on relatively small clique sizes [17, 26, 27, 29], or they look at restricted classes of potentials defined on superpixels or similar predefined image regions [13, 15]. Despite their strongly increase expressive power, these models still mainly encode *local* image properties. Notable exceptions include [20], which shows how a parametric family of region shapes can act as a global prior; and [12], which by extending the cut cost while taking submodularity into account, can deal with challenging situations, such as severe shrinkage and images with shading.

In this work we study truly global topological image properties, such as the *connectedness* of a label region, or the presence and absence of *holes*. In contrast to, *e.g.*, *active contour models* and *level-set methods* [10, 19], little prior work exists on including topology into random field models. To our knowledge the only existing approaches are [25, 30, 31, 21, 28, 11], which we discuss in Section 1.1. Our proposed solution differs significantly from these.

We make three main contributions in this work:

1) we introduce a *minimum perturbation topological prediction (MPTP)* that enforces topology properties by selectively

Figure 1. *Image segmentation with topological side information:* GraphCut-based image segmentation models tend to make topological mistakes, because we only have access to local cues (middle). Enforcing a priori known topological properties (here: connectedness and holefreeness) improves the segmentation (right).

perturbing the energy function,

2) we prove that MPTP is equivalent to previous NP-hard approaches when used with an L^p measure of distortion for $p < \infty$, but that it is *efficiently solvable* for $p = \infty$,

3) we derive an iterative algorithm for *random field image segmentation with topological constraints* that is a) more general and b) more efficient than previous approaches.

1.1. Related Work

To our knowledge, there have been only a few prior attempts to automatically incorporate topological properties[1].

Zeng et al. [31] introduce *TopologyCuts*, a *topology preserving* variant of GraphCut that is inspired by level-set methods. After initialization with a coarse pre-segmentation of correct topology, TopologyCuts iteratively minimizes the energy function while preserving the topology, converging to a local minimum of the energy.

Vicente et al. [30] aim at finding globally optimal solutions to the problem of energy minimization with the topological constraint of a connected foreground region. Because they can prove that even for energy functions with only unary terms this is an NP-hard problem, they restrict their studies to the reduced case of enforcing connectivity between manually selected seed points, which allows, *e.g.*, the interactive segmentation of thin elongated image structures. The authors introduce *DijkstraGC*, a dynamic programming algorithm that yields globally optimal solutions for unary-only energy functions and still provides an approximate solution for energy functions that include pairwise terms.

The only successful approach to tackle the problem of enforcing connectivity of the entire foreground region without manual initialization or seed points is due to Nowozin and Lampert [25]. They also study the problem of enforcing connectivity of the foreground region. They devise an algorithm based on a linear programming (LP) relaxation of the original problem which is solvable to global optimality. However, the relaxation is not tight, such that the resulting image labeling is also not guaranteed to be optimal, and fractional instead of binary valued solutions can occur. Furthermore, the authors found the LP relaxations to not scale well in the number of nodes to label: even with optimized and parallelized code it is not possible to handle more than a few hundred output nodes in reasonable time. We note [21, 28], both of which are closely related to the aforementioned methods.

Finally, it is worth mentioning [11], in which Jain et al. use topology as a measure of the similarity between two different segmentations, specifically in a learning procedure. The algorithm, however, cannot itself guarantee a segmentation with the correct topology.

2. Segmentation with Topological Constraints

We first fix the notation used in the rest of the manuscript. Let $\mathcal{V}$ denote the set of nodes that we want to label. Typically, these are the pixels or superpixels of an image. Each node can take labels from a label set $\mathcal{L}$, *i.e.* a labeling y is an element of $\mathcal{V}^{\mathcal{L}} =: \mathcal{Y}$. For simplicity we only consider binary segmentations, *i.e.* $\mathcal{L} = \{0, 1\}$, where 1 denotes *foreground* and 0 denotes *background*. We will discuss extensions in Section 5.

[1]The term *topological properties* occurs in earlier work, but with a different meaning. For example in [4], the authors use it for the possibility of adding hard constraints for individual pixels to force them in either the foreground or the background region.

Adopting an energy-based random field model, each node $i \in \mathcal{V}$ is equipped with a *unary potential*, $\theta_i(l) \in \mathbb{R}$ for every $l \in \mathcal{L}$, that expresses the cost of assigning the label l to the node i. To simplify the notation we introduce the potential difference

$$\mu_i := \theta_i(1) - \theta_i(0). \tag{1}$$

If necessary, we can compute unary terms $\hat{\theta}_i(y_i) = y_i \mu_i$ from μ_i that result in an energy function equivalent to the original one.

Additionally *pairwise potentials*, $\theta_{ii'}(l, l')$, express the cost of choosing a label combination (l, l') for two nodes (i, i'). Using a graph representation, all non-zero potentials of this kind define an edge set $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$, and typically this is based on the 4- or 8-neighborhood relation. Again for simplicity we will assume the most popular class of symmetric and submodular potentials, in which $\theta_{ii'}(0,0) = \theta_{ii'}(1,1) = 0$ and $\theta_{ii'}(0,1) = \theta_{ii'}(1,0) =: \eta_{ii'}$ for $\eta_{ii'} \geq 0$. We do not consider higher order potentials at this place, but discuss them in Section 5.

The *energy* of a labeling (or segmentation) y is defined as the sum of the above components, which in our case is

$$E(y) := \sum_{i \in \mathcal{V}} \mu_i \, y_i + \sum_{i,i' \in \mathcal{E}} \eta_{ii'} [\![y_i \neq y_{i'}]\!]. \tag{2}$$

The task of *prediction* (image segmentation or restoration) consists of finding the labeling of *minimal energy*:

$$y^* := \operatorname{argmin}_{y \in \mathcal{Y}} E(y). \tag{3}$$

If no other conditions are imposed, Equation (3) can be solved efficiently, for example, using the *GraphCut* algorithm [5, 9].

2.1. Encoding topological properties

For any binary segmentation $y \in \mathcal{Y}$, let $\operatorname{fg}(y) := \{i \in \mathcal{V} : y_i = 1\}$ be its *foreground region*. We express topology properties of $\operatorname{fg}(y)$ by predicates TP_{β_0,β_1}, where β_0 and β_1 upper bound the number of zero-dimensional and one-dimensional topological features, *i.e.*, the number of *connected components* and *holes*[2] [24]. In particular, $TP_{1,\infty}$ expresses if the foreground of a segmentation is connected, and $TP_{\infty,0}$ if it is holefree. $TP_{1,0}$ expresses both properties simultaneously[3]. When there is no ambiguity, we write $TP(y)$ for $TP_{\beta_0,\beta_1}(y)$.

Given an energy function $E(\,\cdot\,; \mu, \eta) : \mathcal{Y} \to \mathbb{R}$ with unary potential vector μ and pairwise parameters η, and given as a topology property *TP*, we study the following two problems.

Problem 1 (Topologically constrained energy minimization – TCEM)**.** *Solve the constrained optimization problem*

$$\operatorname{argmin}_{y \in \mathcal{Y}} E(y) \quad \textit{subject to } TP(y) = \textit{True}. \tag{4}$$

Clearly, if Equation (4) has a feasible solution it will fulfill *TP*. This has been used by previous approaches to perform image segmentation with topological side constraints, see our discussion in Section 1.1. There it has also been observed that Problem 1 is generally NP-hard (*e.g.* [30]), and standard energy minimization techniques, even those designed to handle higher order potential functions, are not directly applicable anymore.

We therefore propose an alternative way for incorporating the information provided by *TP* into an energy minimization framework and we will later show that it is computationally advantageous.

Problem 2 (Minimal perturbation topological prediction – MPTP)**.** *For a given* $1 \leq p \leq \infty$*, solve*

$$\min_{\hat{\mu}} \|\hat{\mu} - \mu\|_p \quad \textit{subject to } TP(y^*) = \textit{True} \tag{5}$$

where $y^* = \operatorname{argmin}_{y \in \mathcal{Y}} E(y; \hat{\mu}, \eta)$.

[2] In our definition of components and holes we always use 4-connectivity of the foreground and 8-connectivity for the background (See [10] for the reason).

[3] More specific criteria based on user input are possible, *e.g.* *"Connect these two components!"*. It will become clear from the construction in Section 3 that our method would be able to handle these as well.

As above, the solution y^* will fulfill *TP* if it exists, and in the remainder of this paper we will show that minimal perturbation topological prediction is a powerful framework for image segmentation with topological constraints.

Intuitively, Problem 2 expresses the idea of *perturbing the unary potentials* of the energy function until an ordinary energy minimization (*i.e.* without additional constraints) yields a solution with the desired properties. Such a step is natural in a computer vision scenario, since there unary potentials are typically only estimated anyway, based, *e.g.*, on local image properties. When faced with a segmentation result that does not have the desired property *TP*, it makes sense modify the energy in a way that the segmentation results better fits our a priori knowledge.

From a practical point of view, the main advantage of MPTP is that the step of modifying E can be decoupled from the step of actual minimizing the energy. We are free to find $\hat{\mu}$ any way we want, as long as it solves Equation (5) and leads to a segmentation fulfilling *TP*. Once we constructed $\hat{\mu}$, we can rely on an arbitrary existing energy minimization technique to perform the actual prediction.

Clearly, a key question in minimal perturbation topological prediction is under which conditions solving Problem 2 is actually easier than the NP-hard Problem 1. Looking at the simplest possible case of a random field with only unary potentials, we can establish the following equivalence result.

Theorem 1. *For an energy function E_μ with only unary terms, Problem 1 is equivalent to Problem 2 with $p = 1$.*

Proof. The minimizing segmentation y^* for E_μ is given by $y_i^* := [\![\mu_i \leq 0]\!]$, consequently, $\mu_i y_i^* = \min(\mu_i, 0)$. For any segmentation y, it follows that

$$\begin{aligned} E_\mu(y) - E_\mu(y^*) &= \sum\nolimits_{\{i:y_i \neq y_i^*\}} \max(\mu_i, 0) - \min(\mu_i, 0) \\ &= \sum\nolimits_{\{i:y_i \neq y_i^*\}} |\mu_i|. \end{aligned} \tag{6}$$

For the same y, we construct a perturbation $\hat{\mu}$ of μ that is closest to μ in L^p norm, and fulfills $y = \operatorname{argmin} E_{\hat{\mu}}(y)$, namely,

$$\hat{\mu}_i := \begin{cases} -\epsilon & \text{if } y_i = 1 \wedge y_i^* = 0, \\ \epsilon & \text{if } y_i = 0 \wedge y_i^* = 1, \\ \mu_i & \text{otherwise.} \end{cases} \tag{7}$$

for any $\epsilon > 0$ and arbitrarily small. The resulting perturbation is

$$\|\mu - \hat{\mu}\|_1 = \sum\nolimits_{\{i:y_i \neq y_i^*\}} |\mu_i| + O(\epsilon). \tag{8}$$

Since this holds for arbitrarily small ϵ, the objective values of Problem 1 and Problem 2 differ only by a constant $E_\mu(y^*)$. Consequently, an optimization over all y with $TP(y)$ will yield the same solution in both cases. □

Corollary 1. *Problem 2 with any $1 \leq p < \infty$ is NP-hard.*

Proof. For $p = 1$, this is a direct consequence of previous theorem, because NP-hardness of Problem 1 in this situation has been shown in [30, Theorem 1]. For $1 < p < \infty$, we derive a reduction from the case of 1-norm. Given an instance of 1-norm, with unary terms μ, we construct a case of p-norm, with unary terms $\mu'_i := sign(\mu_i) \sqrt[p]{|\mu_i|}$. Note that in the absence of pairwise terms, the minimizers of the two energies $E(\cdot; \mu', 0)$ and $E(\cdot; \mu, 0)$ are the same. Using the construction (7) to form $\hat{\mu}'$ from μ' (and suppressing ϵ in the notation), we obtain $\|\hat{\mu}' - \mu'\|_p^p = \sum_{y_i \neq y_i^*} |\mu'_i|^p = \sum_{y_i \neq y_i^*} |\mu_i|$. Therefore, any algorithm for solving the case $1 < p < \infty$ can be used to solve $p = 1$ and polynomially solvability of Problem 2 for any $1 < p < \infty$ implies the same for $p = 1$. □

Interestingly, for $p = \infty$ the situation is fundamentally different, and we will concentrate on this case for the rest of the paper. In the following section we introduce an algorithm for performing image segmentation with topology constraints by solving *minimum perturbation topological prediction* with $p = \infty$. [4]

[4] Note that in the context of topological image properties, $p = \infty$ is in fact a natural choice. For example, when connecting two components, the penalization $\|\hat{\mu} - \mu\|_1$ will be (roughly) proportional to the length of the connecting path, whereas $\|\hat{\mu} - \mu\|_\infty$ expresses the *strongest* obstacle along the path. Properties of such kind are typical for topological studies, because they are invariant under arbitrary continuous reparameterizations of the image plane.

3. The Algorithm

In this section we explain our algorithm for solving Problem 2. Since it relies on tools from *computational topology*, we start by introducing the most relevant terms from that area.

Given a function $\phi : \mathbb{R}^2 \to \mathbb{R}$, we study topology features of a sublevel set (an *object*), $O = \phi^{-1}(\infty, 0] := \{x \in \mathbb{R}^2 \mid \phi(x) \leq 0\}$ (see Figure 2.) Intuitively, given a topology feature α of O (a *component* or a *hole*), its *robustness* [2] is the minimal error we can tolerate to get a function which approximates ϕ and eliminates α, formally, $\rho_\phi(\alpha) = \min_{\alpha \notin \hat{\phi}^{-1}(-\infty,0]} \|\hat{\phi} - \phi\|_\infty$. Different topology features of an object have different robustness. Specifically, it is proved in [2] that

Theorem 2 (Robustness)**.** *For each topology feature, α, there exists a unique pair of points (c_α, d_α) such that $\rho_\phi(\alpha) = \min\{|\phi(c_\alpha)|, |\phi(d_\alpha)|\}$.*

c_α is called *creator* and d_α is called *destroyer* of α. Both are *critical points* of ϕ [23] [5]. Note that the *creator* of a component is a minimal point and the *destroyer* of a component is a saddle point. In Figure 2 left, the the creators of the three components are m_0, m_2 and m_3. The destroyers are s_∞, s_2 and s_3, where s_∞ is a virtual vertex at infinity. There are other critical points, which may not be associated. For the case of holes, the creators and destroyers are saddle points and maximal points respectively. An algorithm to compute (c_α, d_α), and thus $\rho_\phi(\alpha)$, is devised in [2]. Robustness was used to enforce topological constraints in curve and surface evolusion [6]. However, existing results do not provide us with the perturbation to actually eliminate relevant topology features, yet.

3.1. TopoSimp: remove topology noise of an object

Given a function ϕ, and β_0, we devise an algorithm which computes a closest perturbation, $\hat{\phi}$, so that $\hat{\phi}^{-1}(-\infty, 0]$ has no more than β_0 connected components. Other algorithms have been designed to simplify topological structure, but they serve different purposes [8, 1].

We will construct $\hat{\phi}$ by adjust function values of ϕ such that all superfluous components of $O = \phi^{-1}(-\infty, 0]$ are eliminated. We first compute the robustness of all components of O using the algorithm from [2]. We keep the β_0 most robust components and eliminate the rest. For each component α, the creator c_α corresponds to a minimal point and the destroyer d_α corresponds to a saddle. We have two cases depending on which of them has a function value closer to zero.

Case 1, REMOVE: If $|\phi(c_\alpha)| \leq |\phi(d_\alpha)|$, take the connected component containing c_α. For each point in such component, raise its function value to $0 + \epsilon$.

Case 2, MERGE: If $|\phi(c_\alpha)| > |\phi(d_\alpha)|$, take a path which goes through d_α, connects α and another component α' with a bigger robustness. For each point in such path with positive function value, decrease its function value to $0 - \epsilon$. Note that this path is more special than described. It is the path in the component tree, as frequently used in mathematical morphology [22]. Any point in such path has the value $\leq \phi(d_\alpha)$. More details will be provided in the appendix.

Here we let ϵ be an arbitrarily small positive number. The raising (resp. decreasing) of the value at each point would turn its sign to positive (resp. negative). The first case is equivalent to removing the whole component from the segmentation, when the corresponding minimal point is closer to zero than the saddle point. The second case is equivalent to merging two components by connecting a path between them, so that the highest point of the path is the saddle (see Figure 2).

We can extend the algorithm to include the constraint of an upperbound β_1 of the number of holes. Intuitively, the algorithm ignores the β_1 most robust holes. For the others, it chooses to either merge with other holes, or seal them.

Theorem 3. *Our algorithm computes* $\operatorname{argmin}\|\hat{\phi} - \phi\|_\infty$*, such that* $TP_{\beta_0,\beta_1}(\hat{\phi}^{-1}(-\infty, 0]) = True$.

Proof. In the algorithm, to eliminate a topology feature α the amount of perturbation for each individual point is upperbounded by the minimum of $|\phi(c_\alpha)|$ and $|\phi(d_\alpha)|$, and thus the robustness of α. Therefore, $\|\hat{\phi} - \phi\|_\infty$ is upperbounded by the maximum of the robustness of the $(\beta_0 + 1)$th most robust component and the $(\beta_1 + 1)$th most robust hole. According to Theorem 2, this is the best perturbation we can achieve in order to eliminate all superfluous components and holes. □

Intuition. For each component of the sublevel set (a basin), the algorithm has two potential ways of eliminating it. The first is filling the basin up; the cost is then the distance of the bottom from the ground. The second way involves digging a canal so that the basin is connected to another basin which is harder to eliminate. The cost of digging the canal is the maximal altitude that can be reached along the corresponding path. Depending on which cost is smaller, the algorithm will choose to either fill the basin or dig a canal.

[5] In case if some critical points share a same function value, the pairing could be ambigous. This can be solved by a random choice.

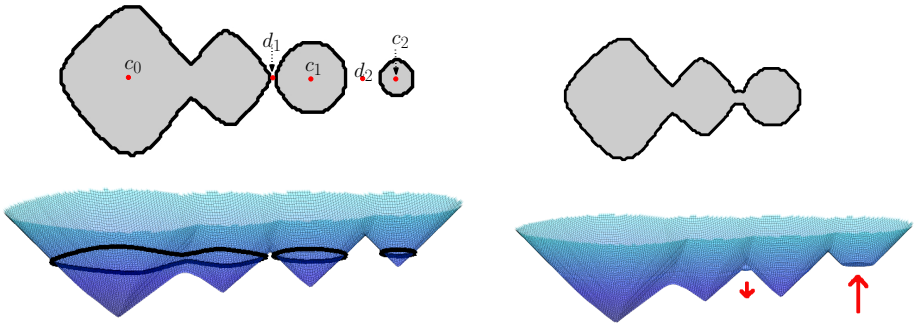


Figure 2. Functions and the corresponding objects (the shaded regions). **Left:** A function whose corresponding object has three components. **Right:** The closest perturbation in which the middle and right components are eliminated by merging and removing respectively. (Images are from [6].)

3.2. Enforcing topology constraints in segmentation

We integrate the aforementioned algorithm, TopoSimp, into the image segmentation framework to solve MPTP (Problem 2). The resulting algorithm we call *TopoCut*.

First, when the binary or higher order term is zero ($\eta = 0$) our algorithm solves MPTP problem by taking μ_i (Equation (1)) as the function ϕ. In such case, the optimal segmentation $y^* = \operatorname{argmin}_{y \in \mathcal{Y}} E(y; \mu, 0)$ labels a pixel i foreground when $\mu_i \leq 0$ and background otherwise. Therefore the foreground of y^* is identical to the sublevel set $\mu^{-1}(-\infty, 0]$. The TopoSimp algorithm computes the closest L^∞ perturbation of μ so that the sublevel set satisfies *TP*, and thus solves the MPTP problem.

In the case when $\eta \neq 0$, we adapt our algorithm to heuristically solve the problem, even though we have no optimality guarantee anymore. The intuition is as follows. We start with an original optimal segmentation y^* and we try to eliminate its components and holes when their numbers are above β_0 and β_1. To do so, we adjust the unary potential using robustness. Next, we apply graph cuts to the new unary potential and the original binary potential. This process is repeated until a segmentation with the correct topology is found.

At each iteration, to elimerate extra topology features, we measure their "robustness" according to the landscape of the unary function μ. Specifically, we construct a new function, ϕ, whose sublevel set $\phi^{-1}(-\infty, 0]$ is the same as the foreground of y^*. Furthermore, within the foreground region (resp. background region), the landscape of ϕ is the same as μ. Let $C = \max |\mu_i|$, we then define ϕ as follows.

$$\phi_i := \begin{cases} \mu_i + C & \text{if } y_i = 0, \\ \mu_i - C & \text{if } y_i = 1. \end{cases}$$

Next, we apply TopoSimp to the constructed function ϕ. The output is a perturbation $\hat{\phi}$ of ϕ. We use such perturbation to adjust the unary term, namely, $\hat{\mu} = \mu + \hat{\phi} - \phi$. This adjustment enforces regions to be background (in the *removing* cases) and paths to be foreground (in the *merging* cases).

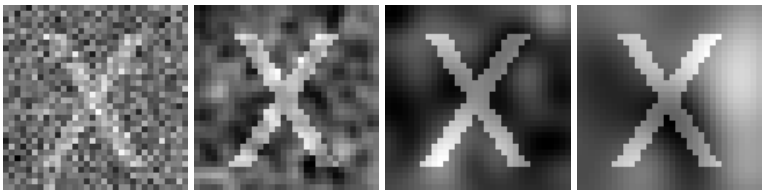
Next, we use the adjusted $\hat{\mu}$ together with the other energy contribution to obtain a new segmentation, $\hat{y}^* = \operatorname{argmin}_{y \in \mathcal{Y}} E(y; \hat{\mu}, \eta)$. If $\hat{y}^*$ still does not satisfy the topology constraint, we use $\hat{y}^*$ and $\hat{\mu}$ to construct another ϕ and we again apply TopoSimp. We proceed iteratively until the segmentation result satisfies *TP*.

Notice that at each iteration, the algorithm would only adjust ϕ towards 0. Therefore, the value $C = \max |\mu_i|$ remains constant.

One justification of the TopoCut algorithm is that for $\eta = 0$ it produces the MPTP segmentation, even though the optimal L^∞ norm of the perturbation is increased by C. Also the critical points associated to each topology feature remain the same, only the absolute values of their function values are all increased by C. The suggested path to merge components, and the suggested components to be removed all remain the same.

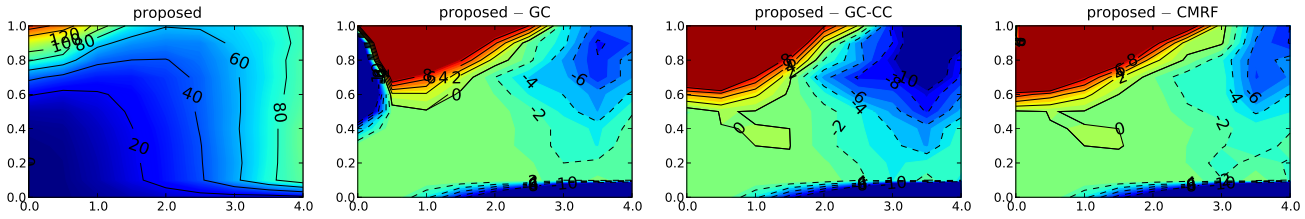
4. Experiments

We evaluate the TopoCut algorithm in experiments on synthetic and natural images. The synthetic setup with known ground truth and noise model allows us to analyze and discuss the properties of MPTP quantitatively. The natural images illustrate how image segmentation can benefit from the the integration of topological constraints.

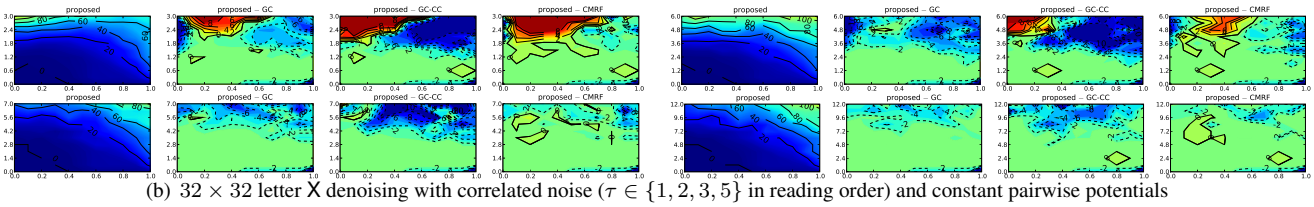


(a) uncorrelated (b) $\tau = 1$ (c) $\tau = 3$ (d) $\tau = 5$

Figure 3. Visualization of unary potentials with different forms of Gaussian noise. τ denotes the correlation strength.



(a) 32×32 letter X denoising with uncorrelated noise and random pairwise potentials



(b) 32×32 letter X denoising with correlated noise ($\tau \in \{1, 2, 3, 5\}$ in reading order) and constant pairwise potentials

Figure 4. Quantitative image reconstruction results (best viewed in color). Column 1: Hamming loss of TopoCuts for varying σ (y-axis) and k (x-axis). Columns 2–4: Error difference to baselines: MRF with connectivity constraints (CMRF) [25], GraphCut (GC), connected component heuristic (GC-CC). Larger version of the plots can be found in the appendix.

Image Denoising. We follow the setup introduced in [16] that subsequently was used in [25] to quantitatively evaluate *topologically constrained energy minimization*. We compute unary potentials for a 32×32 pixel grid by disturbing a predefined X-shaped ground truth labeling y^{GT} with independent Gaussian noise,$\mu_i = 1 - 2y_i^{GT} + \mathcal{N}(0, \sigma)$. Pairwise potentials are chosen by connecting each pixel to its 4-neighborhood and sampling submodular Potts potentials $\eta_{ii'} = |\mathcal{N}(0, k/2)|[\![y_i \neq y_{i'}]\!]$ (the factor $1/2$ occurs only to make our parametrization consistent with previous work).

We vary $\sigma \in \{0, 0.1, \ldots, 1\}$ and $k \in \{0, 0.5, 1, \ldots, 4\}$ to obtain random fields with different amounts of signal noise and smoothness priors, see Figure 3(a) for an example image. For each parameter setting we sample 30 random field instances, and we perform random field prediction using different methods. Because we know a priori that the ground truth is connected and holefree shape, we use TopoCuts with $TP_{1,0}$ as topological property. We compare to the connectivity enforcing approach of [25] (CMRF) using C++ implementation provided on the author's homepage. In addition we include two baselines: energy minimization using a GraphCut without topological constraints (GC), and the heuristic of segmenting with GraphCut and enforcing foreground connectivity by keeping only the largest connected component of the resulting foreground region (GC-CC). The approaches [30, 31] are not applicable in this setup, because they require manual initialization or user-provided seed points. We measure the quality of the resulting reconstructions by their Hamming distance to the known ground truth. Figure 4(a) reports averaged numeric results over the 30 runs.

The experimental setup described above was originally devised to measure the performance of generic energy minimization approaches, and it does not reflect well the situation how computer vision application today typically make use random fields. In particular, the choice of random values for the pairwise potential terms, and the assumption of uncorrelated noise in the unary potentials are unrealistic. Consequently we also performed experiments using random fields that better reflect the situation of image segmentation: we choose Potts pairwise potentials with uniform weight $\eta_{ii'} = k[\![y_i = y_{i'}]\!]$ and sample the unary potentials $\mu = (\mu_i)_{i \in \mathcal{V}}$ from a *correlated* noise distribution, $\mu = 2y^{GT} - 1 + \sigma\epsilon$ where $\epsilon \sim \mathcal{N}(0, \Sigma_\tau)$ are τ-correlated Gaussian noise images that we obtain by smoothing independent Gaussian noise with a Gaussian of variance τ^2 over the pixel grid. Figures 3(b)–3(d) show example images. We repeat the experimental setup described above with these choices. The results are summarized in Figure 4(b).

Discussion of Results. Figure 4 shows that knowledge of the target topology, *i.e.* connectedness and holefreeness in

method	err (global w) [%]	err (per-image w) [%]
TopoCuts	7.0 ± 0.7	6.1 ± 0.6
GC	7.5 ± 0.7	6.5 ± 0.6

Table 1. Segmentation accuracy on GrabCut image dataset.

our example, can indeed improve the image restoration results. For large regions of the parameter set, TopoCuts improves reconstruction results over the GC segmentation, as well as over the GC-CC heuristic (blue and blue-green regions). Only if very strong and uncorrelated noise is added to the unaries, perturbation based topological prediction yields worse results than the baseline methods (red regions). This effect can be understood by the fact that strong uncorrelated noise leads a large number of small (typically single-pixel) foreground components in the GraphCut segmentation. If their unary potential is strong enough, TopoCuts connect these to the main contour, introducing erroneous additional foreground pixels in the process. CMRF is less prone to this effect, because its L^1 criterion suppresses small components even of high unary values if they lie far away from the main contour.

For medium and strongly correlated noise (and this is common for natural images) our algorithm outperforms the baselines in terms of reconstruction accuracy, and it is generally comparable to CMRF or even better. This makes sense, because correlated noise is less likely to create single pixel outliers of very strong unary potential. Instead, more homogeneous of larger size components occur, and for these the L^∞ criterion is as reliable as the L^1 one.

An advantage of TopoCuts is that it always creates binary segmentations, whereas CMRF's output can have fractional entries, because of the underlying LP-relaxation approach. In our experiments, this happened for 13% of the images, where on average 2.7% of pixels were fractionally labeled. In order to obtain a binary reconstruction we treat all nodes with fractional values as part of the foreground. The alternative of rounding to the nearest integer would not ensure connectivity, and it also resulted in slightly worse reconstruction accuracy during our experiments.

Object Segmentation. To show the potential of TopoCuts also for practical segmentation tasks, we applied it to the *GrabCut* dataset of natural images, following the procedure introduced in [3]. For each of the 50 images in the dataset, we use the provided *lasso*-trimap to compute unary potentials for foreground and background using a color mixture model. We set the pairwise potentials on an 8-neighborhood using the contrast sensitive scaling of [3]. With the resulting energy function we perform figure-ground segmentation using GraphCut (GC) and our proposed algorithm (TopoCuts), using foreground 4-connectivity as topological property[6]. The energy function has one free parameter, the weighting w of unary to pairwise terms. We report results for choosing $w \in \{0, 0.2, \ldots, 1.4\}$ by two selection criteria: setting w globally to the value that leads to lowest total segmentation error over all image *(global)*, and setting w on a per-image basis to the value that leads to minimal total error *(per image)*. The segmentation quality is measured separately for each image by the total number of mislabeled pixels divided by the number of pixels in the uncertainty region as specified by the tripmap. Table 4 reports the results in form of mean and standard error of the mean over the 50 images. It shows that including topological information improves the averaged segmentation scores, but only slightly so. This is not surprising, given that for many of GrabCut images the GraphCut segmentation is already connected, and both methods yield the same result in these cases.

The advantage of topological side information becomes apparent when looking at those examples where the GC segmentation does not have the correct topology, such that TopoCuts modified the solution.

Figure 5 shows illustrative examples, in particular demonstrating how the smoothness prior from pairwise potentials and topological constraints complement each other. If w is chosen too small ($w = 0$ in the figure), the GC segmentation creates many small components. TopoCuts discards some of these, but it also connects many of them to the foreground region by at least a foreground path of single pixel width. With increasing weights for the pairwise terms, most noise components are discarded by the GraphCut. However, significant parts of the image remain unconnected. TopoCuts enforces paths between these, and because of the contrast sensitive edge weights, the paths are become larger image segments during the subsequent energy minimization. With too highly weighted, the pairwise terms (*e.g.* Figure 5, top left with $w = 0.8$), also suppress relevant components during the GraphCut. Based on the MPTP objective, TopoCuts is not able to recover these.

Runtime. For the 32×32 images TopoCuts requires only fractions of a second for the topological segmentation, and the total runtime is in the range of one second, mainly due to data input and output. For larger images, the iterative calls to the GraphCut dominate the overall runtime algorithm: if the original GraphCut is already of the correct topology, TopoCuts terminates immediately. Otherwise, typically less 5 iterations are necessary and TopoCuts runs at not less then 20% the speed of an unconstrained GraphCut.

In contrast, CMRF is computationally very demanding, and it does not scale beyond very small images size. Unless the initial GraphCut segmentation was already connected (in which case it converges almost immediately), CMRF required

[6] All object in the GraphCut dataset are connected, but some exhibit holes. Therefore we do not also enforce holefreeness.

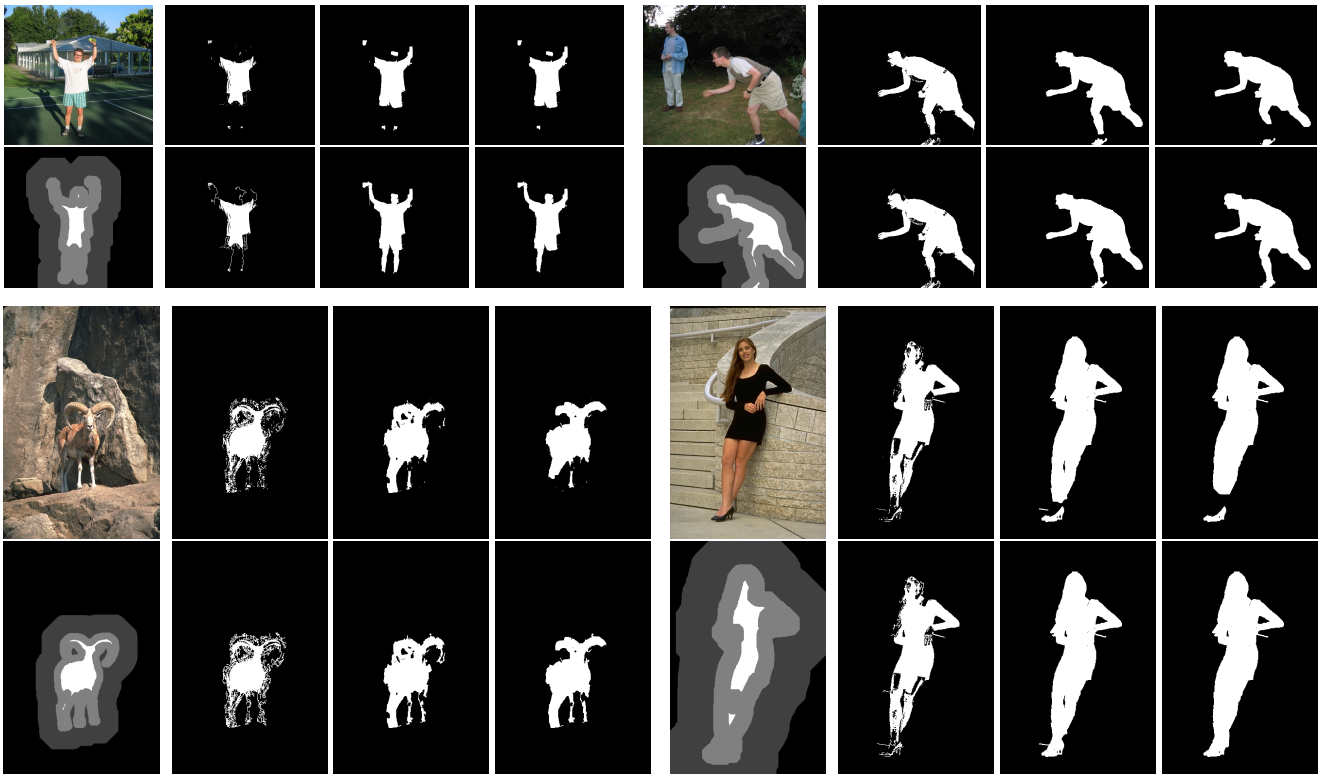


Figure 5. Example segmentations from of GrabCut dataset. Each group shows (in reading order): original image, GraphCut segmentation with $w \in \{0, 0.4, 0.8\}$, trimap, TopoCuts segmentation with w as above and *TP* = *"*fg(y) *is connected"*.

on average 131 seconds to process one 32×32 images. In the most difficult cases (strong noise, no pairwise potentials), segmenting a single image sometimes took over three hours of CPU time.

5. Extensions and Outlook

In this work we chose to explain perturbation-based topological prediction using binary image segmentations only and with an energy free of higher order terms. However, we are very positive that the concepts described will carry over to more general situations. In *multi-label segmentation*, we can compute and influence topological properties of each label region. *Higher order potentials* can be integrated into the iterative segmentation algorithm the same way as we do it for the pairwise terms: we ignore them during TopoSimp, but include them during the actual energy minimization steps. Topological properties make particular sense for *images of higher dimension*, such as three-dimensional medical image volumes where anatomical knowledge provides topological side information. The mathematical tool to handle this situation are already available. However, the TopoSimp would need to be extended to handle the richer topology of $\mathbb{R}^3$. Finally, the proof of Theorem 1 shows us that the *minimal perturbation topological prediction* does not have to be restricted to topological properties, but could be extended also for other problems of energy minimization with additional constraints. It is an interesting open question which other image properties in general lend themselves to the minimal perturbation idea.

Acknowledgments

The first author is supported by the Austrian Science Fund (FWF) grant No. P20134-N13. The authors would like to thank Sebastian Nowozin for helpful discussions.

References

[1] D. Attali, M. Glisse, S. Hornus, F. Lazarus, and D. Morozov. Persistence-sensitive simplification of functions on surfaces in linear time. In *Workshop on Topology-based Methods in Data Analysis and Visualization*, 2009. 5

[2] P. Bendich, H. Edelsbrunner, D. Morozov, and A. Patel. The robustness of level sets. In *European Symposium on Algorithms*, 2010. 5

[3] A. Blake, C. Rother, M. Brown, P. Perez, and P. Torr. Interactive image segmentation using an adaptive gmmrf model. In *ECCV*, 2004. 8

[4] Y. Boykov and G. Funka-Lea. Graph cuts and efficient nd image segmentation. *IJCV*, 70(2):109–131, 2006. 2

[5] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimizaton in vision. *PAMI*, 26(9):1124–1137, 2004. 3

[6] C. Chen and D. Freedman. Topology Noise Removal for Curve and Surface Evolution. *Medical Computer Vision. Recognition Techniques and Applications in Medical Imaging*, pages 31–42, 2011. 5, 6

[7] H. Edelsbrunner and J. Harer. *Computational topology: an introduction*. American Mathematical Society, Providence, RI, 2010. 11

[8] H. Edelsbrunner, D. Morozov, and V. Pascucci. Persistence-sensitive simplification functions on 2-manifolds. In *ACM Symposium on Computational Geometry*, 2006. 5

[9] D. M. Greig, B. T. Porteous, and A. H. Seheult. Exact maximum a posteriori estimation for binary images. *Journal of the Royal Statistical Society (B)*, 51(2):271–279, 1989. 1, 3

[10] X. Han, C. Xu, and J. Prince. A topology preserving level set method for geometric deformable models. *PAMI*, 25(6):755–768, 2003. 1, 3

[11] V. Jain, B. Bollmann, M. Richardson, D. Berger, M. Helmstaedter, K. Briggman, W. Denk, J. Bowden, J. Mendenhall, W. Abraham, et al. Boundary learning by optimization with topological constraints. In *CVPR*, 2010. 1, 2

[12] S. Jegelka and J. Bilmes. Submodularity beyond submodular energies: coupling edges in graph cuts. In *CVPR*, 2011. 1

[13] T. H. Kim, K. M. Lee, and S. U. Lee. Nonparametric higher-order learning for interactive segmentation. In *CVPR*, 2010. 1

[14] R. Kindermann and J. L. Snell. *Markov random fields and their applications*. American Mathematical Society, Providence, RI, 1980. 1

[15] P. Kohli, L. Ladický, and P. H. S. Torr. Robust higher order potentials for enforcing label consistency. *IJCV*, 82(3):302–324, 2009. 1

[16] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts? *PAMI*, pages 147–159, 2004. 7

[17] N. Komodakis and N. Paragios. Beyond pairwise energies: Efficient optimization for higher-order MRFs. In *CVPR*, 2009. 1

[18] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*, 2001. 1

[19] C. Le Guyader and L. Vese. Self-repelling snakes for topology-preserving segmentation models. *Image Processing, IEEE Transactions on*, 17(5):767–779, 2008. 1

[20] V. Lempitsky, A. Blake, and C. Rother. Image segmentation by branch-and-mincut. In *ECCV*, 2008. 1

[21] V. Lempitsky, P. Kohli, C. Rother, and T. Sharp. Image segmentation with a bounding box prior. In *ICCV*, 2010. 1, 2

[22] F. Meyer and S. Beucher. Morphological segmentation. *Journal of Visual Communication and Image Representation*, 1(1):21–46, 1990. 5

[23] J. W. Milnor. *Morse theory*. Princeton Univ Press, 1963. 5

[24] J. R. Munkres. *Elements of algebraic topology*. Westview Press, 1984. 3

[25] S. Nowozin and C. H. Lampert. Global connectivity potentials for random field models. In *CVPR*, 2009. 1, 2, 7

[26] B. Potetz and T. S. Lee. Efficient belief propagation for higher-order cliques using linear constraint nodes. *CVIU*, 112(1):39–54, 2008. 1

[27] S. Ramalingam, P. Kohli, K. Alahari, and P. Torr. Exact inference in multi-label CRFs with higher order cliques. In *CVPR*, 2008. 1

[28] C. Rhemann, C. Rother, P. Kohli, and M. Gelautz. A spatially varying PSF-based prior for alpha matting. In *CVPR*, 2010. 1, 2

[29] C. Rother, P. Kohli, W. Feng, and J. Jia. Minimizing sparse higher order energy functions of discrete variables. In *CVPR*, 2009. 1

[30] S. Vicente, V. Kolmogorov, and C. Rother. Graph cut based image segmentation with connectivity priors. In *CVPR*, 2008. 1, 2, 3, 4, 7

[31] Y. Zeng, D. Samaras, W. Chen, and Q. Peng. Topology cuts: A novel min-cut/max-flow algorithm for topology preserving segmentation in n-d images. *CVIU*, 112(1), 2008. 1, 2, 7

Appendix

A. Implementation Details of TopoSimp

In this section, we provide implementation details of the TopoSimp algorithm which is introduced in Section 3.1 of the paper. See [7] for correctness of the algorithm. We will release an open source implementation after acceptance of the paper.

A.1. CompTree: computing component tree, as well as creators and destroyers

Given an image and a function value $\phi(v)$ for each pixel v, we compute a component tree as well as all creators and destroyers associated to each component α of the sublevel set $\phi^{-1}(-\infty, 0]$ as follows.

1. Construct the 4-connected graph of the image, whose vertices are pixels and edges connect neighboring pixels.
2. Assign each edge $e = (v_i, v_j)$ the function value $\phi(e) = \max(\phi(v_i), \phi(v_j))$.
3. Sort all edges in ascending order according to the function value. Call such sequence S.
4. Initialize the component tree T as the set of all vertices.
5. For each edge e in S, we check if $T \cup \{e\}$ remains a tree. If no, skip e. Otherwise,
   - Let T_1 and T_2 be the two subtrees of T connected by e.
   - Let $v_1 = \text{argmin}_{v \in T_1} \phi(v)$, $v_2 = \text{argmin}_{v \in T_2} \phi(v)$. Without loss of generality, assume $\phi(v_1) < \phi(v_2)$.
   - If $\phi(v_2) < 0$ and $\phi(e) > 0$, v_2 and e are the creator and destroyer of a component α.
   - $T \leftarrow T \cup \{e\}$.

Using the union-find data structure, the time complexity of such algorithm is $O(nA(n))$, where n is the number of vertices and $A(n)$ is the inverse of the Ackermann function, which can be assumed to be constant for all practical purposes. Therefore the algorithm is essentially linear time.

A.2. Path for merging components.

When merging one component α into others (CASE 2), we need a path which

- goes through the destroyer of α, d_α;
- connects α with a component α' which is more robust than α;
- any point in the path has a function value $\leq \phi(d_\alpha)$.

This path can be found in the component tree. In specific, given α, the destroyer d_α is an edge in T connecting two subtrees T_1 and T_2 during Step 5 of the CompTree algorithm. Let v_1 and v_2 be the minimal vertices of T_1 and T_2, as defined in CompTree. The path in T connecting v_1 and v_2 is the desired path.

A.3. Extension to eliminating extra holes.

To extend the algorithm for eliminating holes. We need to assign function values to not only vertices and edges as we did in CompTree, but also faces of the grid. For each face, let the function value be the maximal ϕ of its four vertices.

We define a new graph. Vertices of such graph include the dual vertices of all the faces, together with a virtual vertex v_∞, which is dual to the whole region outside the image grid. Edges of such graph include the dual edges of all edges which are not in the component tree T computed by the previous algorithm CompTree. See Figure 6.

For each vertex of this new graph, which is dual to the face f, assign a function value as $-\phi(f)$. Assign $-\infty$ to the virtual vertex v_∞. For each edge, which is dual to e, assign a function value as $-\phi(e)$.

Next, apply CompTree to this graph to build a component tree $\bar{T}$. During this process, the creators and destroyers are associated to holes of the zero sublevel set of the original function. The path in $\bar{T}$ is the path which is used to merge two holes, in case this is needed.

In an actual implementation, it can happen that we have to apply different function adjustments to the same vertex (pixel) in order to eliminate different components or holes. In such case, we always respect the adjustment due to the elimination of

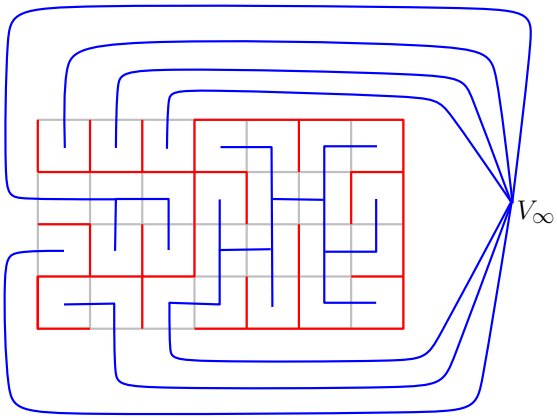
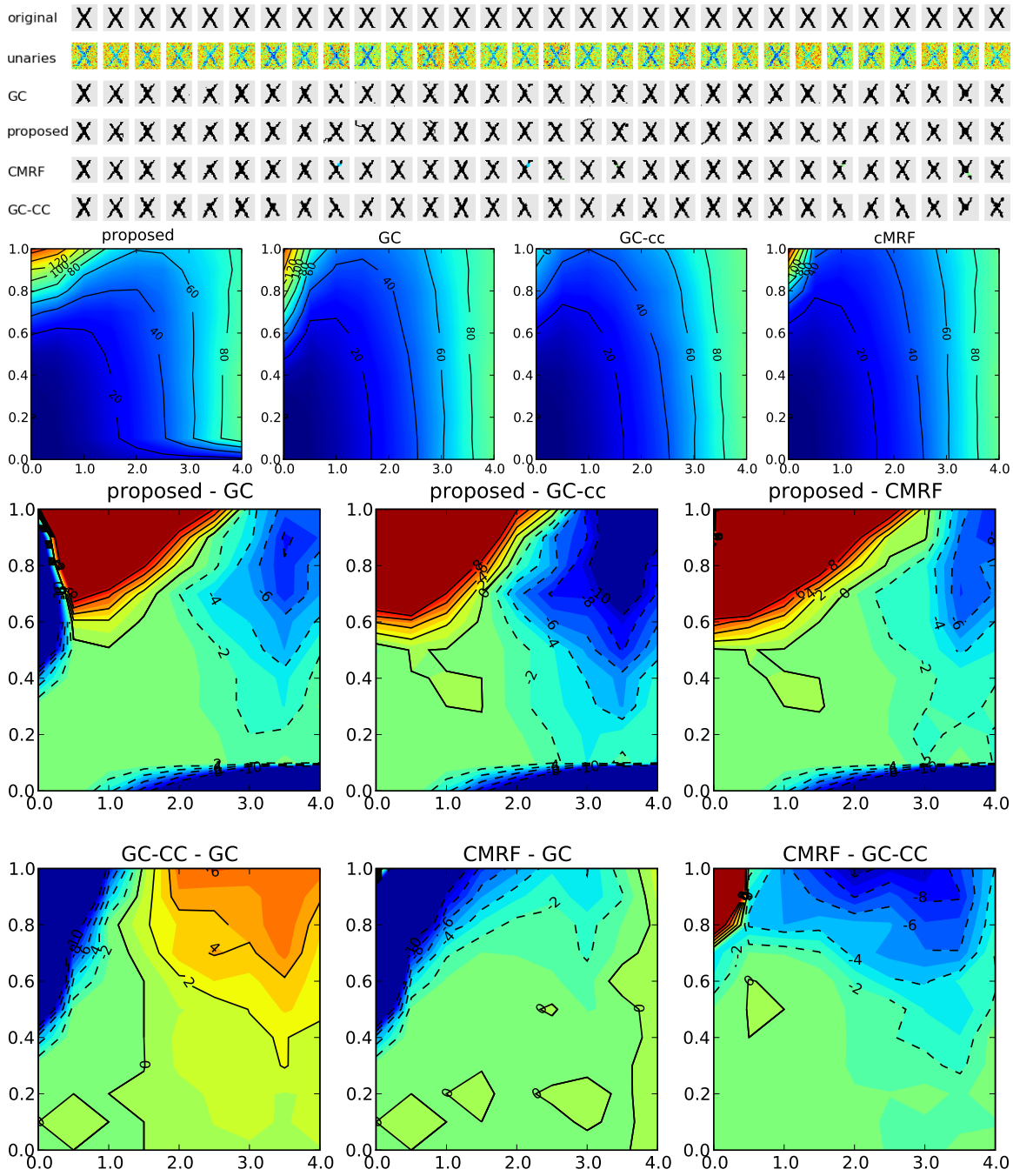


Figure 6. Visualization of the tree constructions. Gray: the 4-connected graph (vertices are pixels). Red: component tree T. Blue: dual component tree $\bar{T}$.

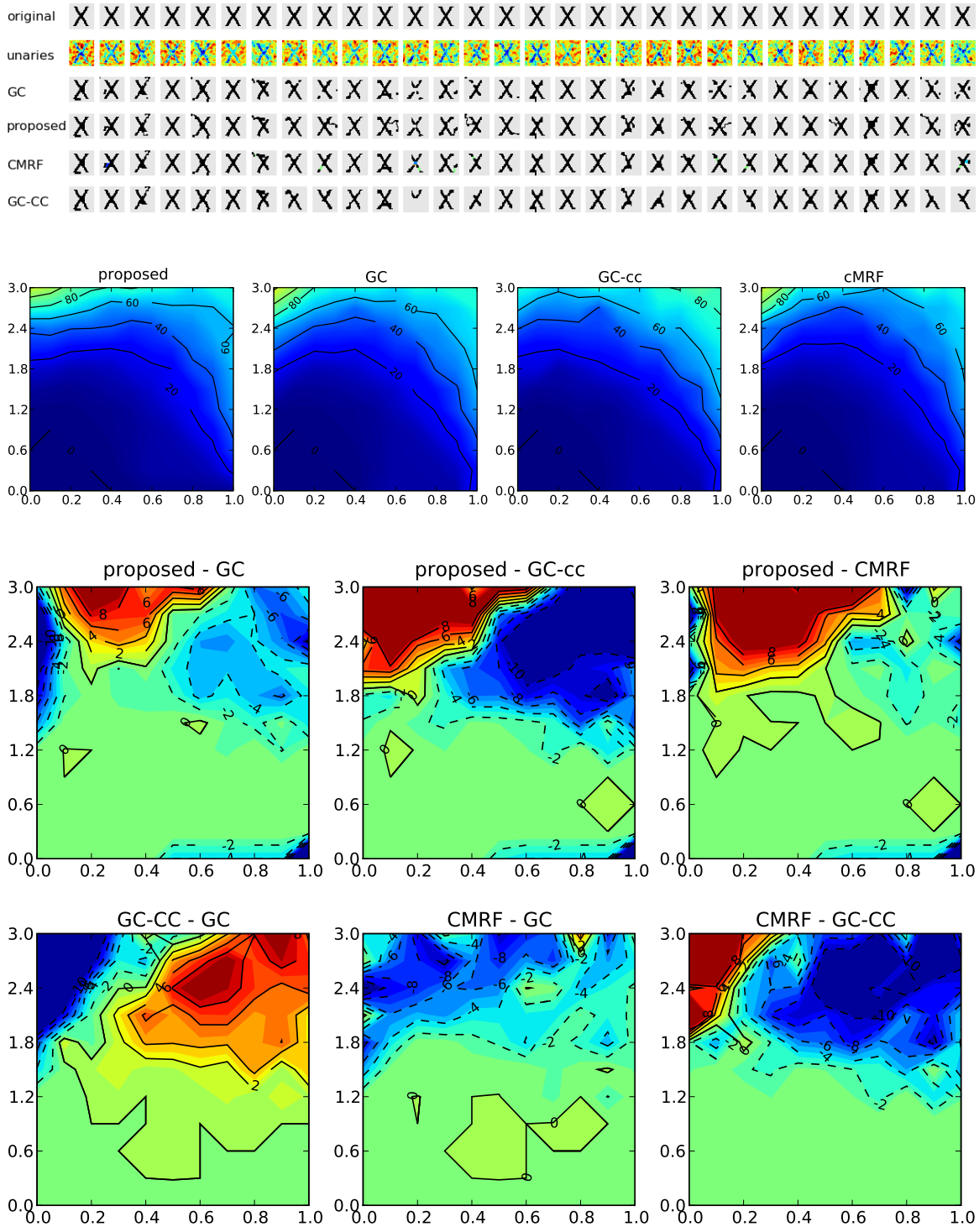
the most robust component/hole. This way, we ensure that the most robust component/hole is eliminated. If an iteration of the algorithm is not sufficient to eliminate all topology noise, we apply the algorithm again. The algorithm always eliminates all topology noise in a linear number of iterations, since it eliminates at least one components/hole per iteration, and it never creates new ones.

B. Quantitative Experiments: Image Reconstruction (Figure 4)

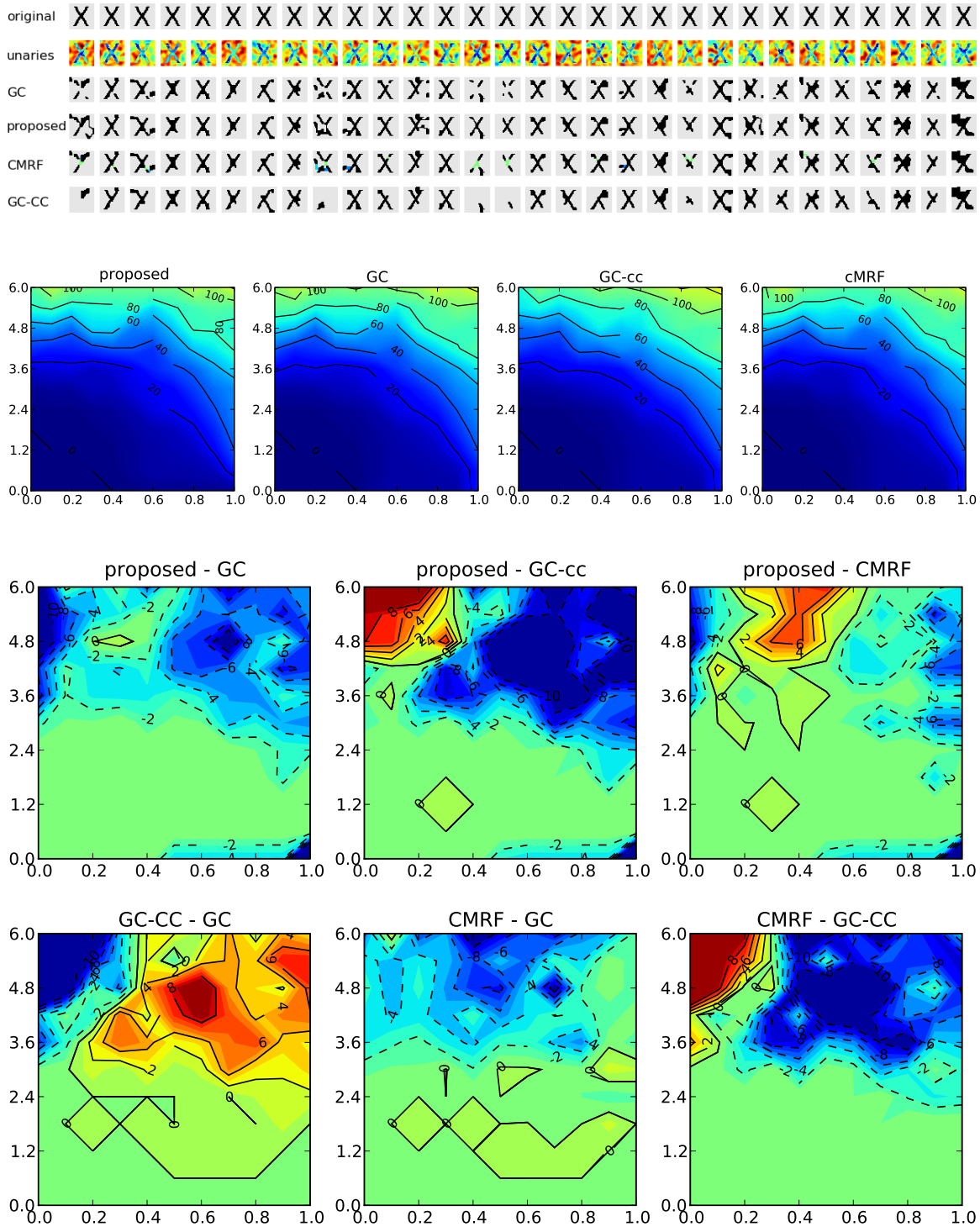
Denoising results for 32×32 letter X with *uncorrelated noise* and *random pairwise potentials*. Top images: example images, unary potentials ($\sigma = 0.8, k = 2.0$) and results of different methods: *TopoCuts* (proposed), *GraphCut (GC)*, *connected component heuristic* (GC-CC), and *MRF with connectivity constraint* (CMRF). Value besides red/blue in CMRF output depict fractional values: light red=$\frac{3}{4}$, green=$\frac{1}{2}$, medium blue=$\frac{1}{3}$, light blue=$\frac{1}{4}$. Middle images: absolute Hamming error for different methods. Bottom images: difference in Hamming error between methods.



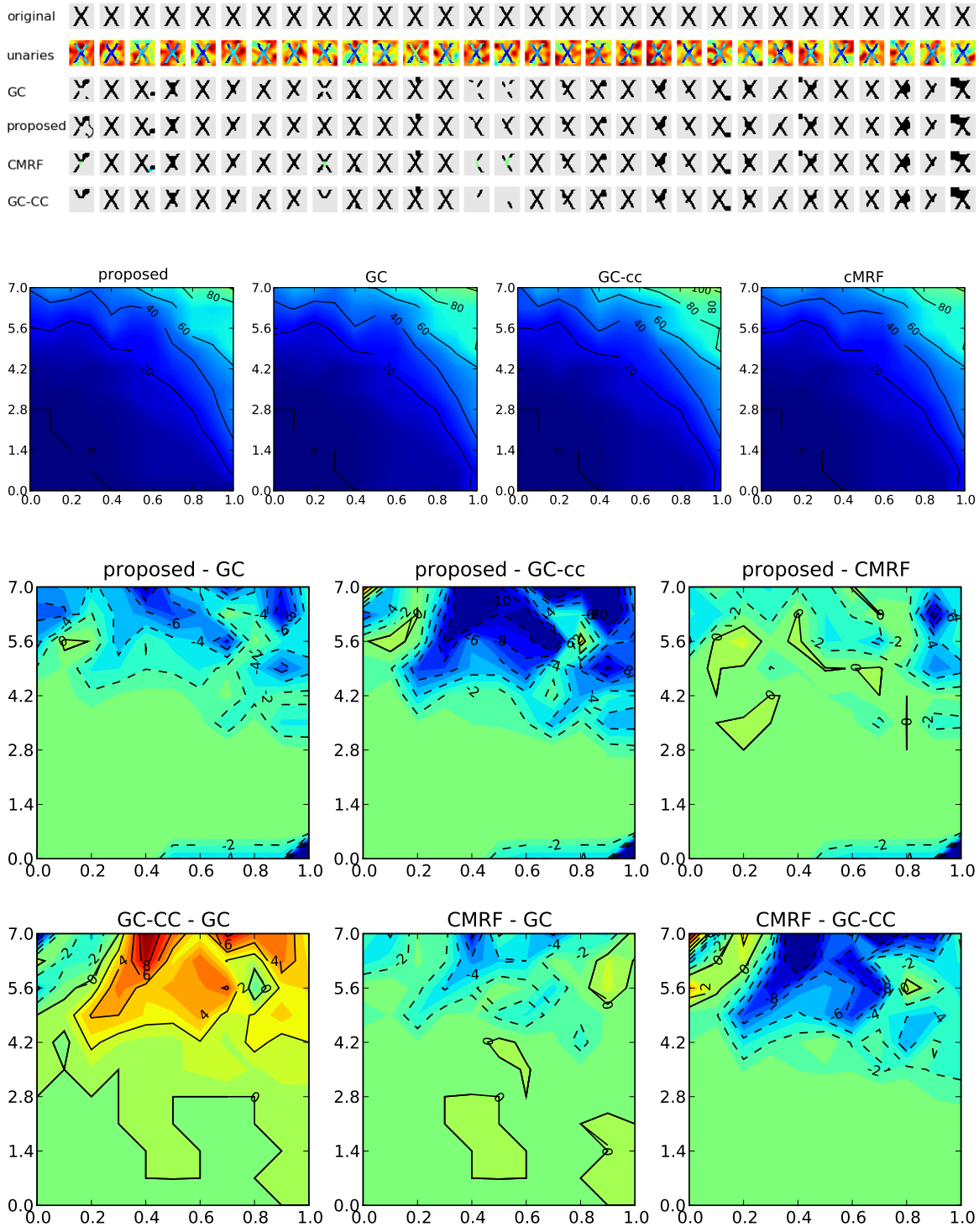
Denoising results for 32×32 letter X with *correlated noise* ($\tau = 1$) and *constant pairwise potentials*. Top images: example images, unary potentials ($\sigma = 2.4, w = 0.4$) and results of different methods.



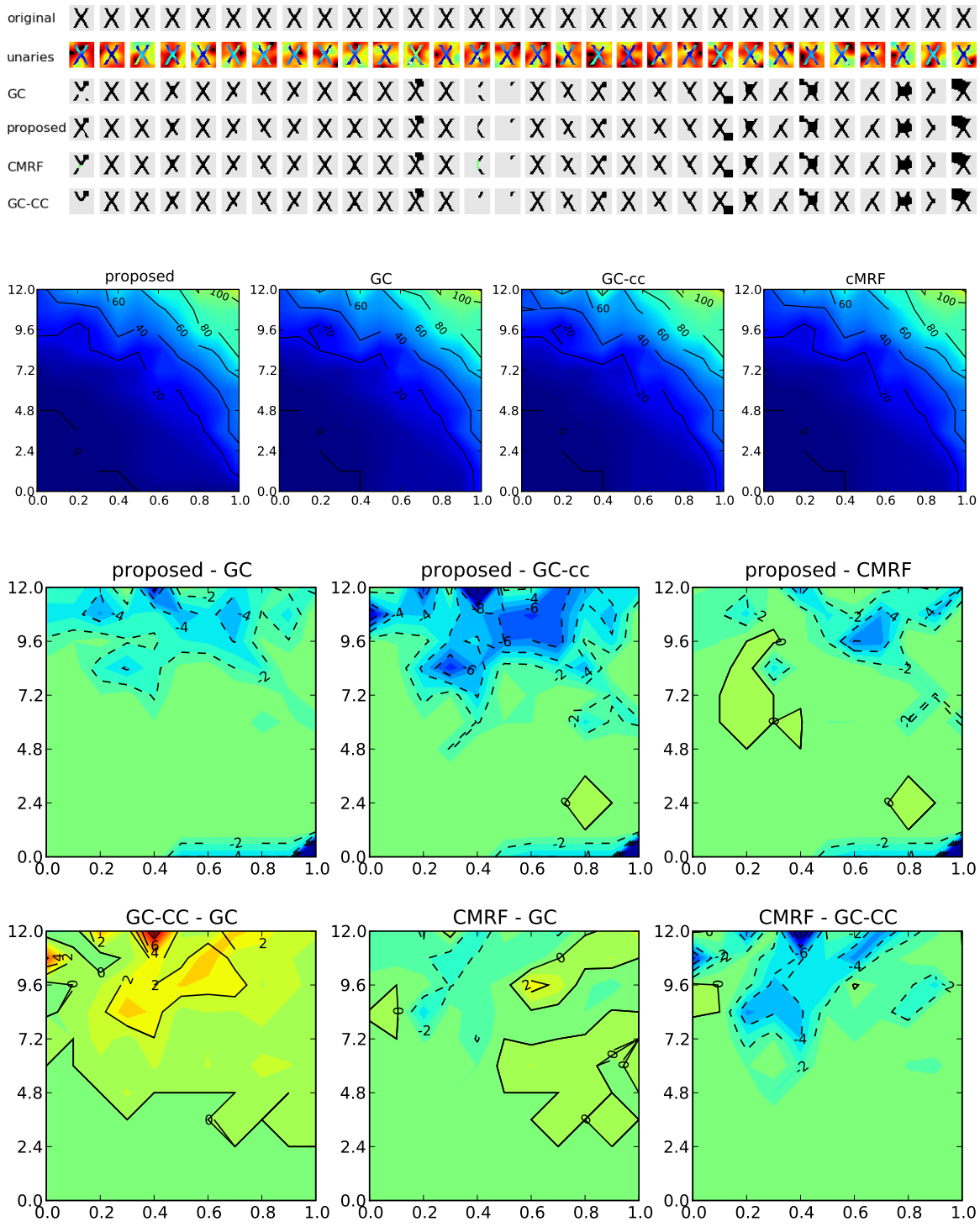
Denoising results for 32×32 letter X with *correlated noise* ($\tau = 2$) and *constant pairwise potentials*. Top images: example images, unary potentials ($\sigma = 4.8, w = 0.4$) and results of different methods.



Denoising results for 32×32 letter X with *correlated noise* ($\tau = 3$) and *constant pairwise potentials*. Top images: example images, unary potentials ($\sigma = 5.6, w = 0.4$) and results of different methods.



Denoising results for 32×32 letter X with *correlated noise* ($\tau = 5$) and *constant pairwise potentials*. Top images: example images, unary potentials ($\sigma = 9.6, w = 0.4$) and results of different methods.



C. Qualitative Results: Image Segmentation (Figure 6)

Segmentation results on *GrabCut* dataset. The images are presented in decreasing order of average overlap difference $1 - \frac{fg^{GC} \cap fg^{TC}}{fg^{GC} \cup fg^{TC}}$ between GraphCut (GC) and TopoCuts (TC) segmentation (*i.e.* with and without topological connectivity constraint).

Top row: original image, provided trimap. Remaining images (in western reading order): segmentation results with $w \in \{0, 0.2, \ldots, 1.4\}$. White regions were selected as foreground by both, GraphCut and TopoCuts. Green regions were added by TopoCuts to the GC segmentation. Red regions were removed by TopoCuts from the GC segmentation.

Image "tennis", average overlap difference 14.78%

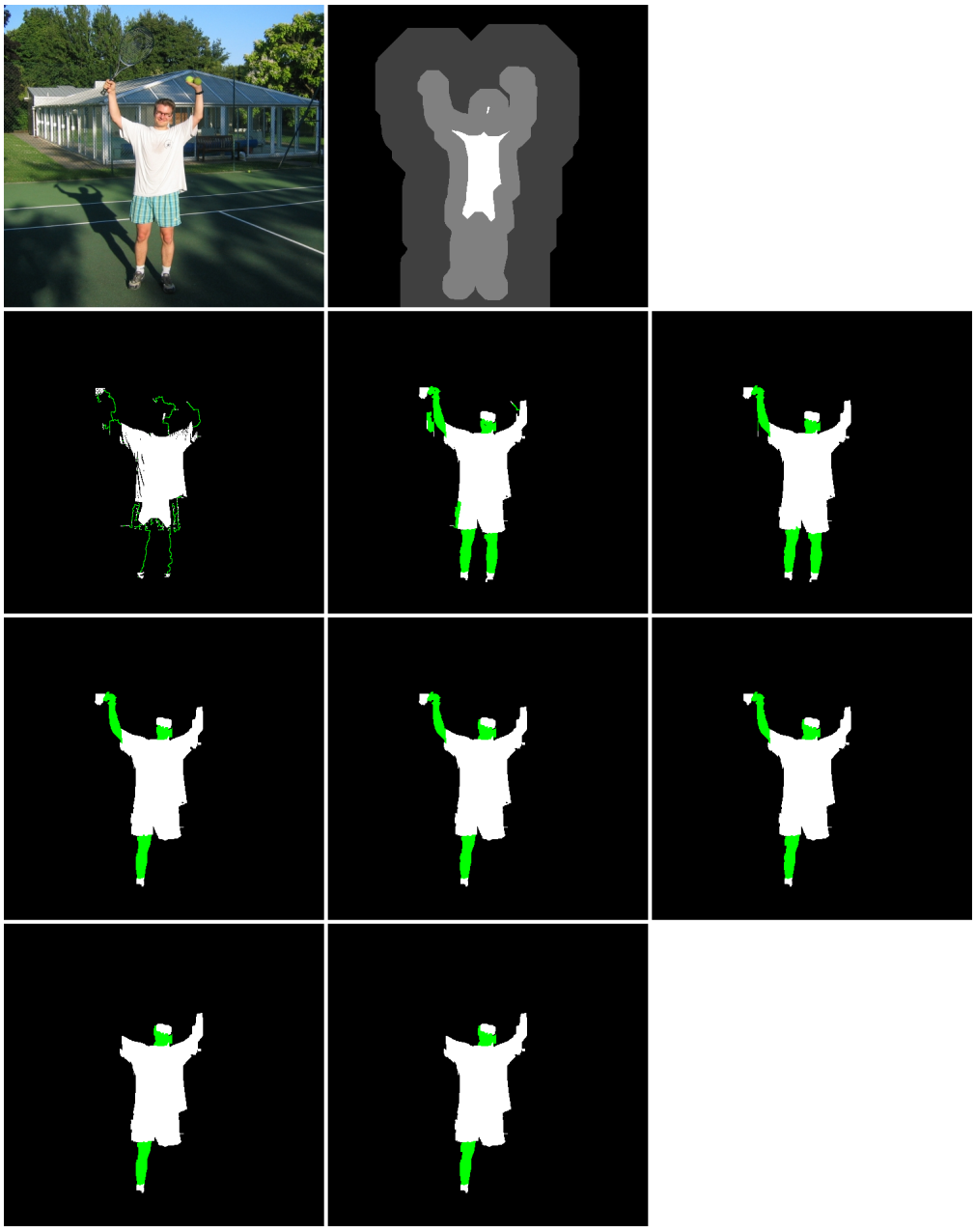


Image ”304074”, average overlap difference 8.60%

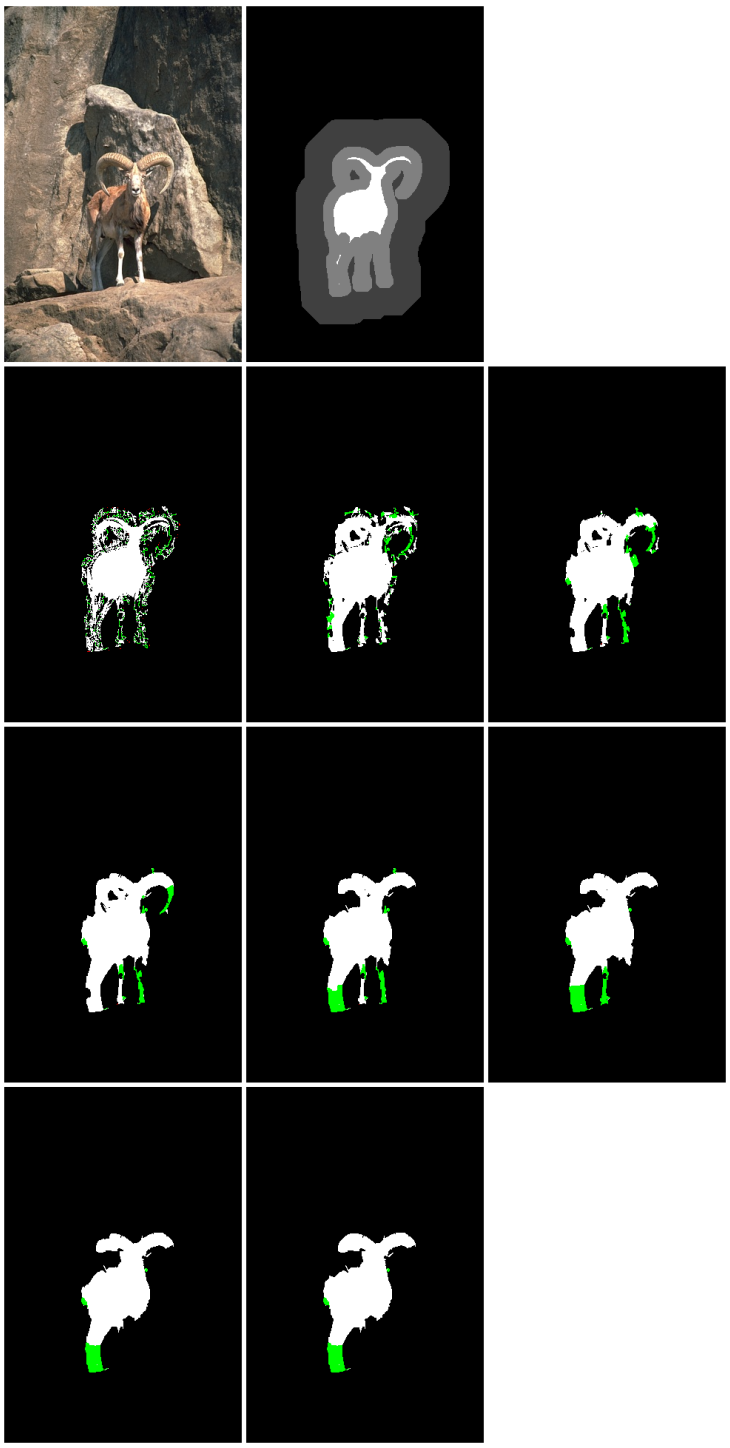


Image ”bush”, average overlap difference 7.19%

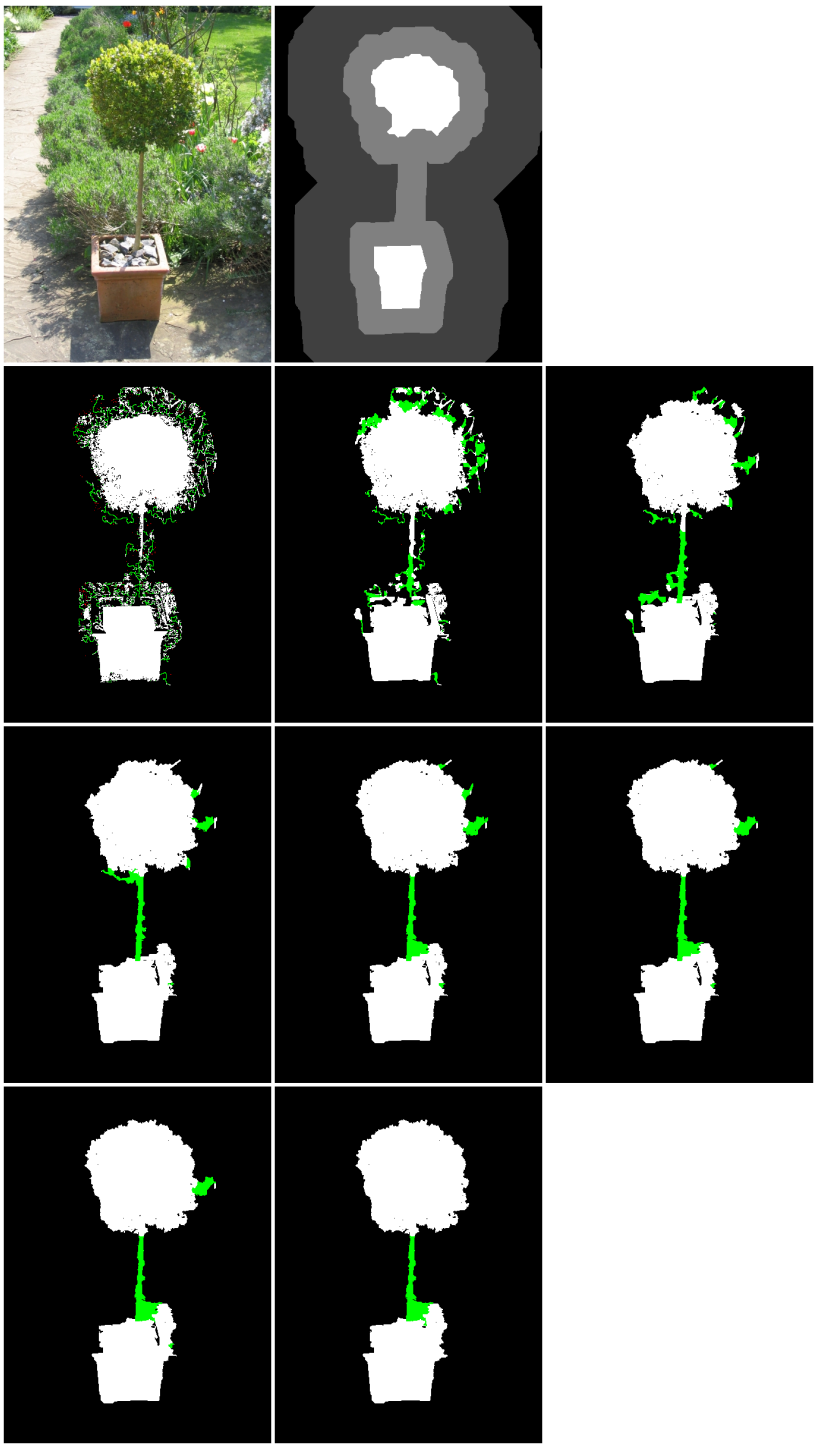


Image ”106024”, average overlap difference 5.10%

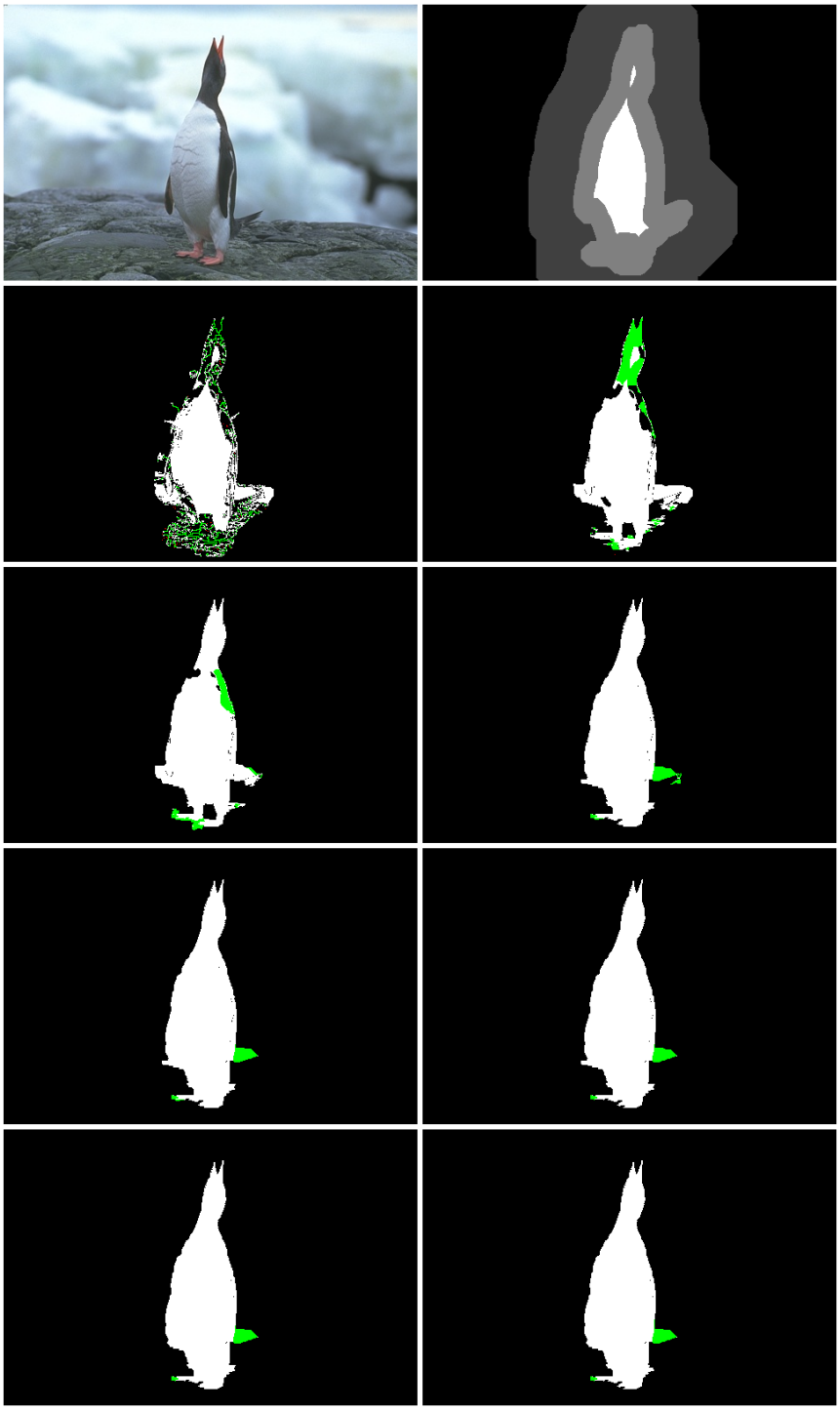


Image ”326038”, average overlap difference 4.73%

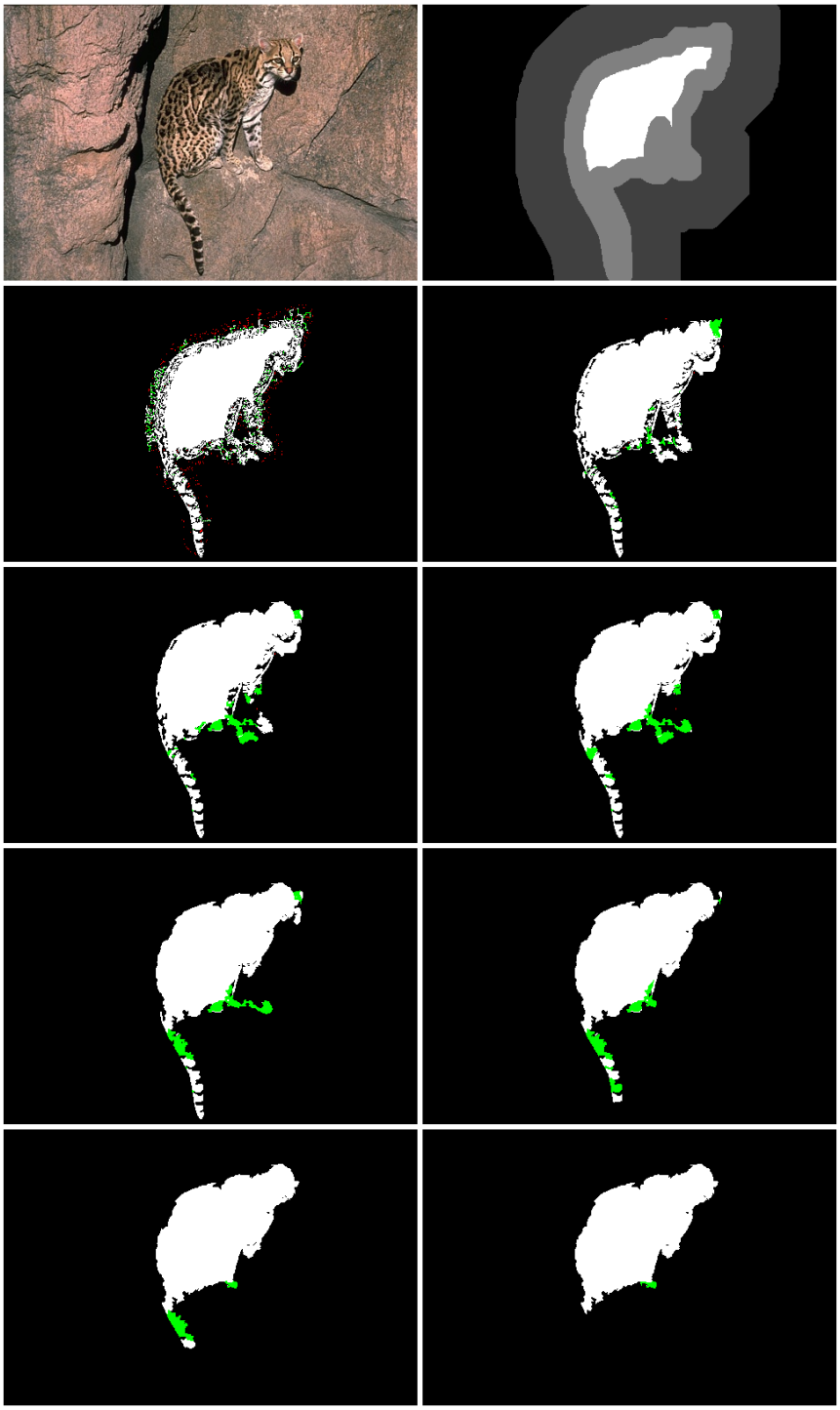


Image ”24077”, average overlap difference 4.56%

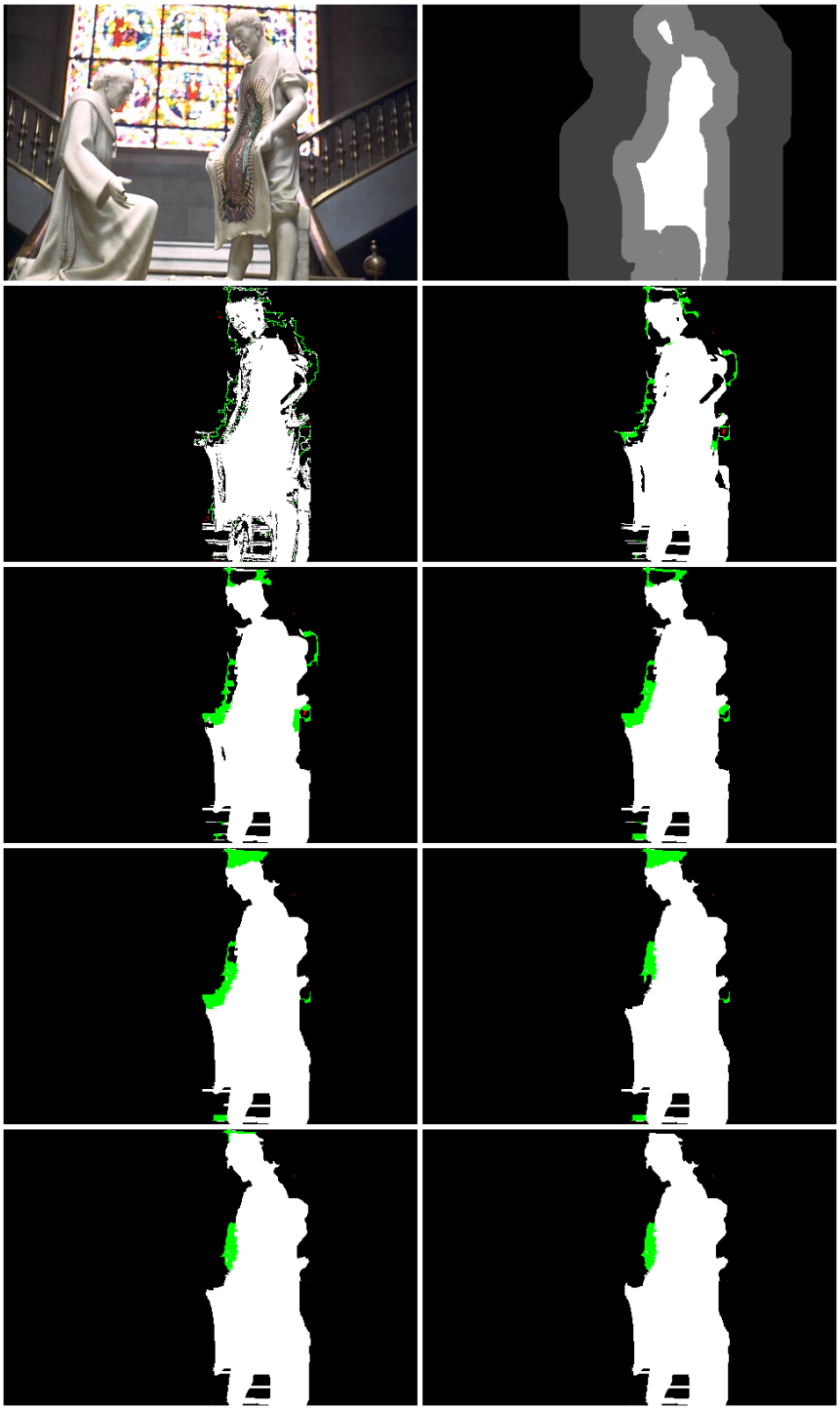


Image ”person5”, average overlap difference 4.46%

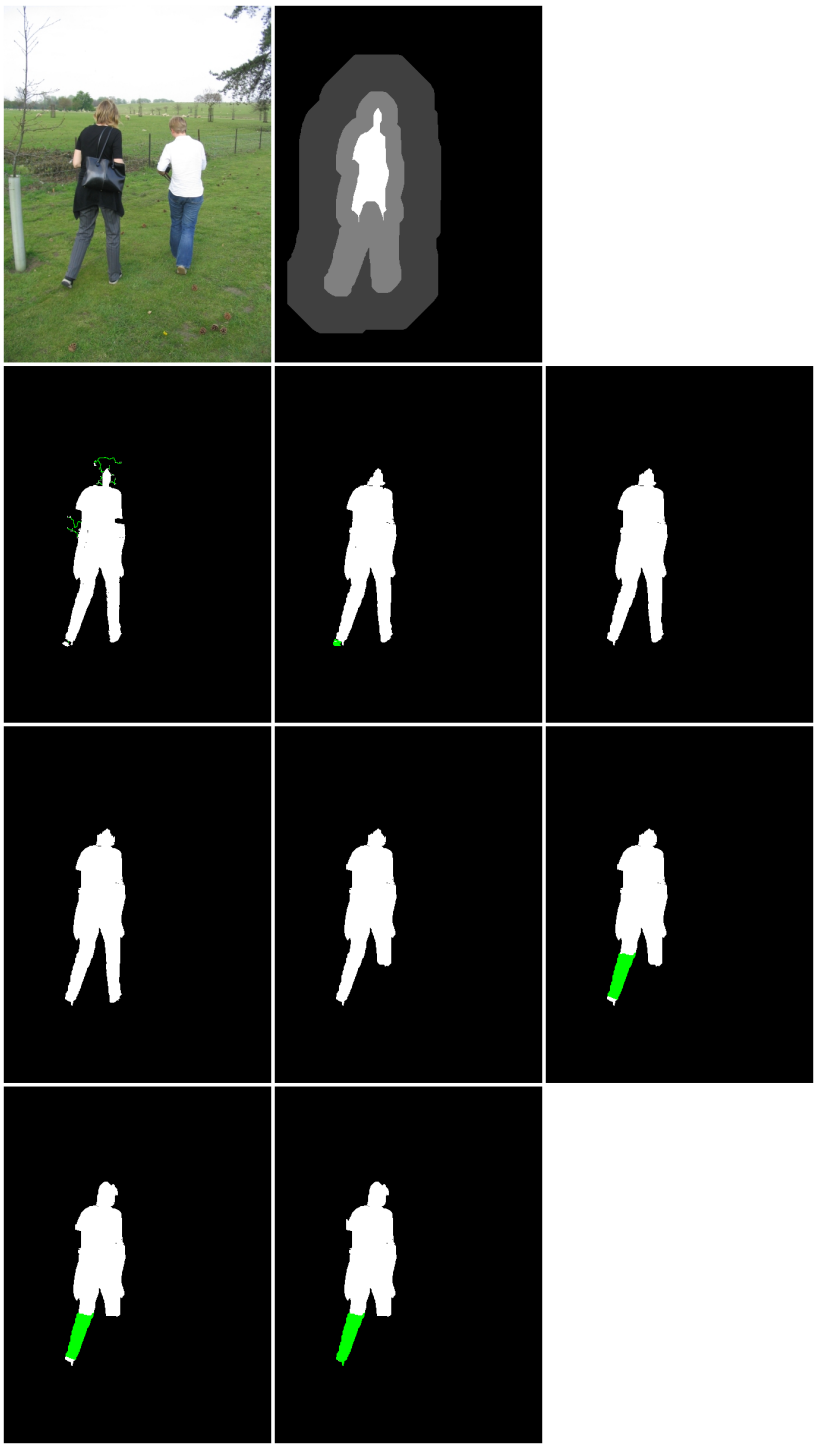


Image ”388016”, average overlap difference 4.30%

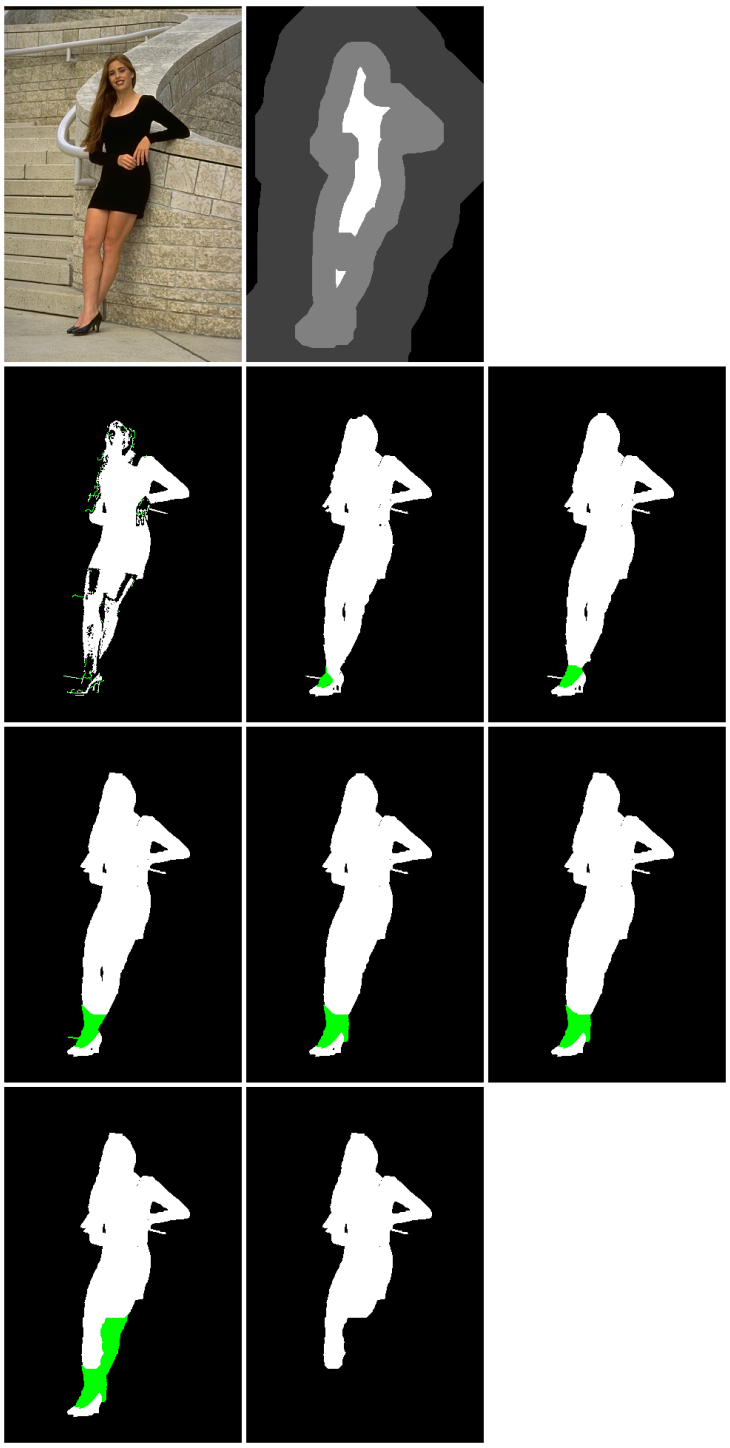


Image ”scissors”, average overlap difference 3.10%

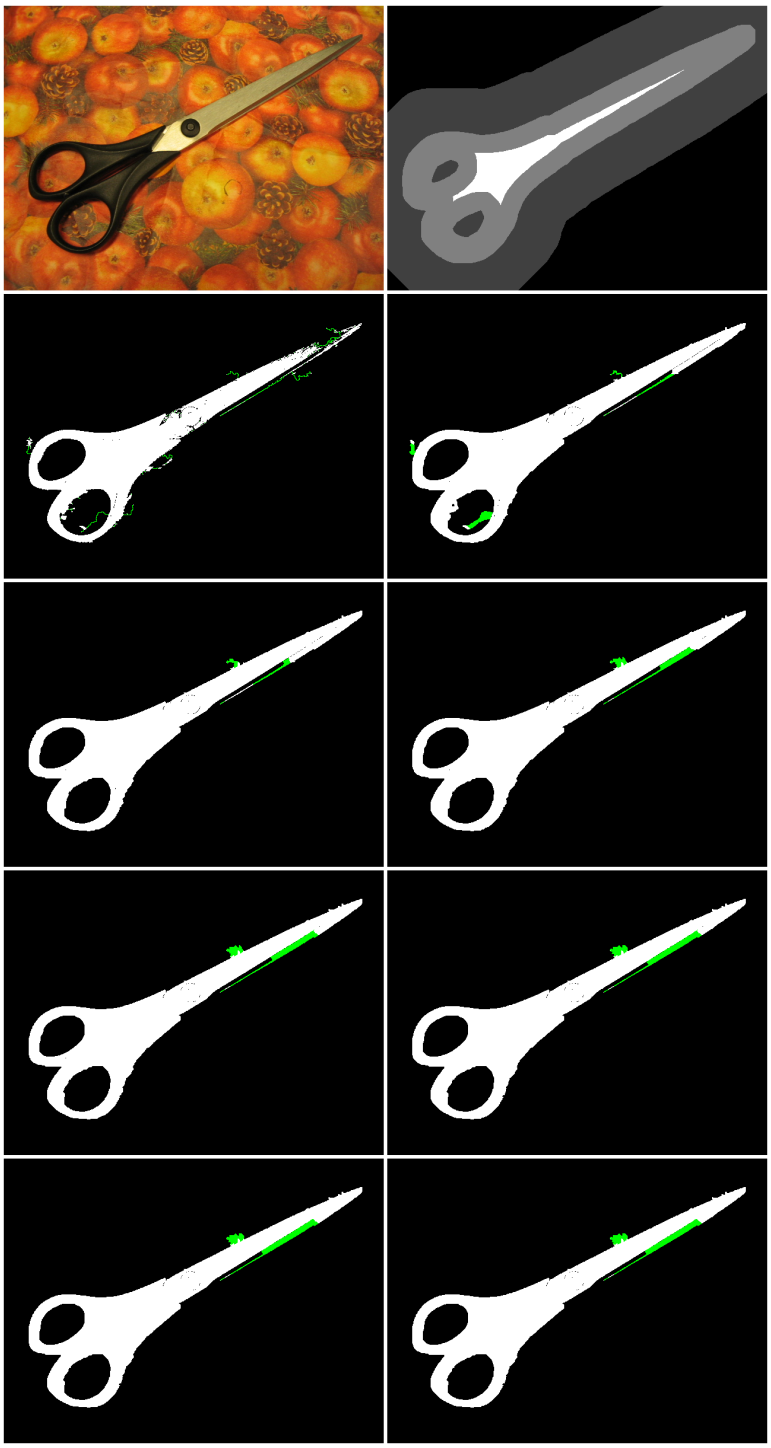


Image ”bool”, average overlap difference 2.57%

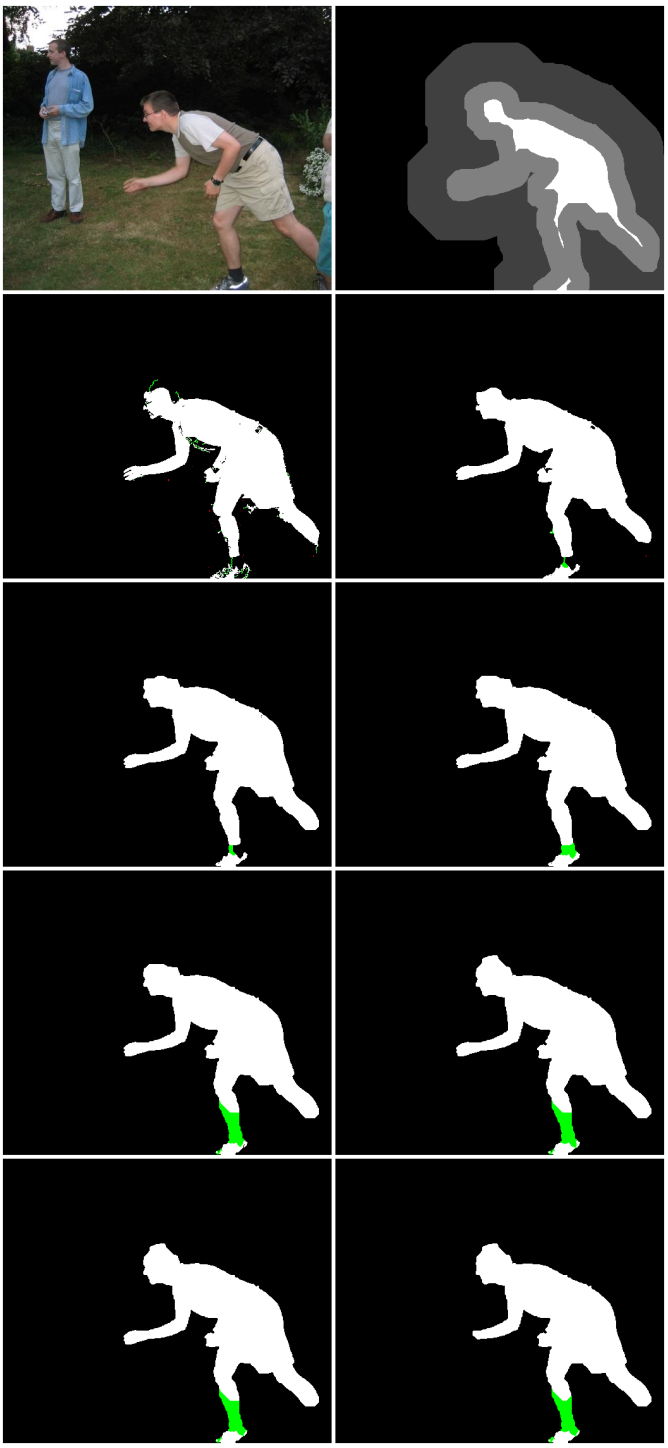


Image ”person4”, average overlap difference 2.34%

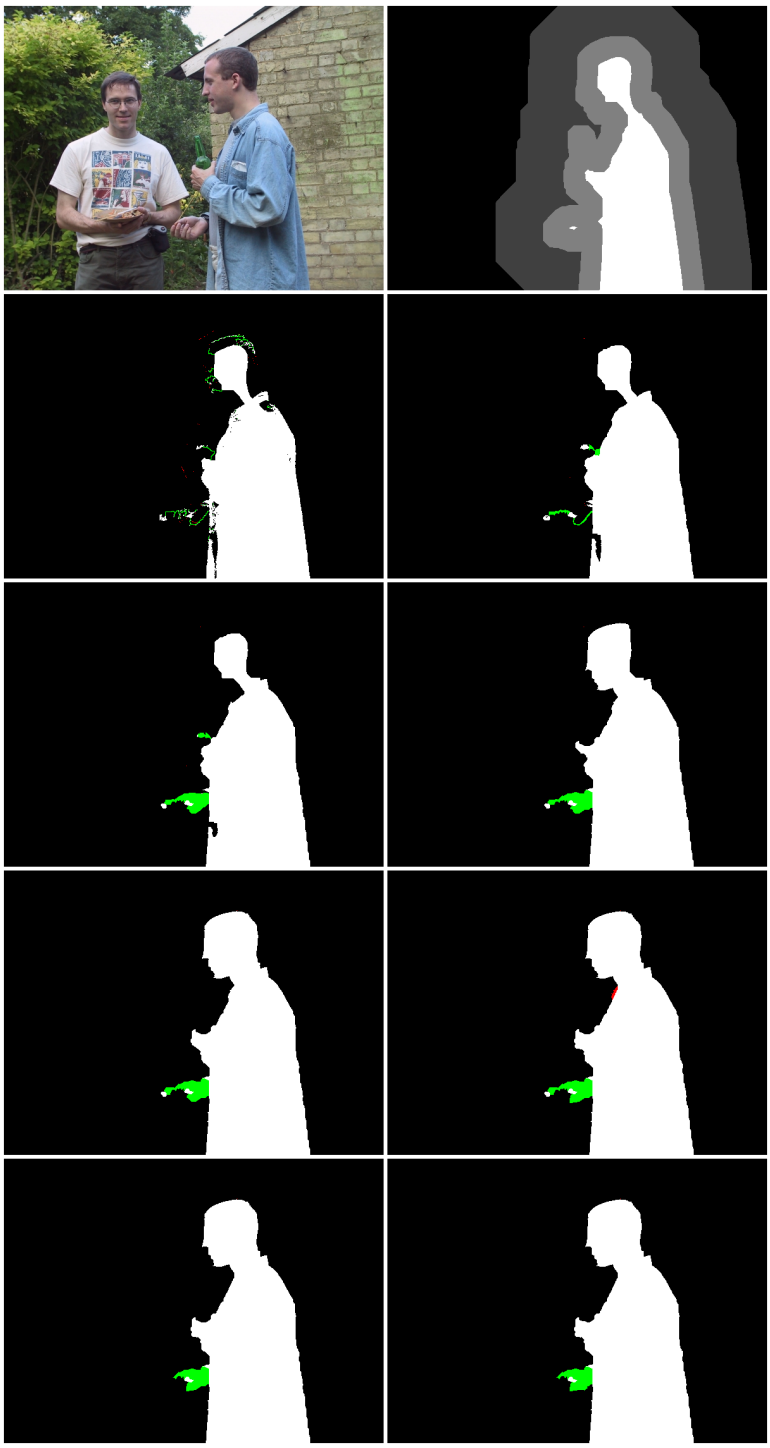


Image ”person7”, average overlap difference 1.62%

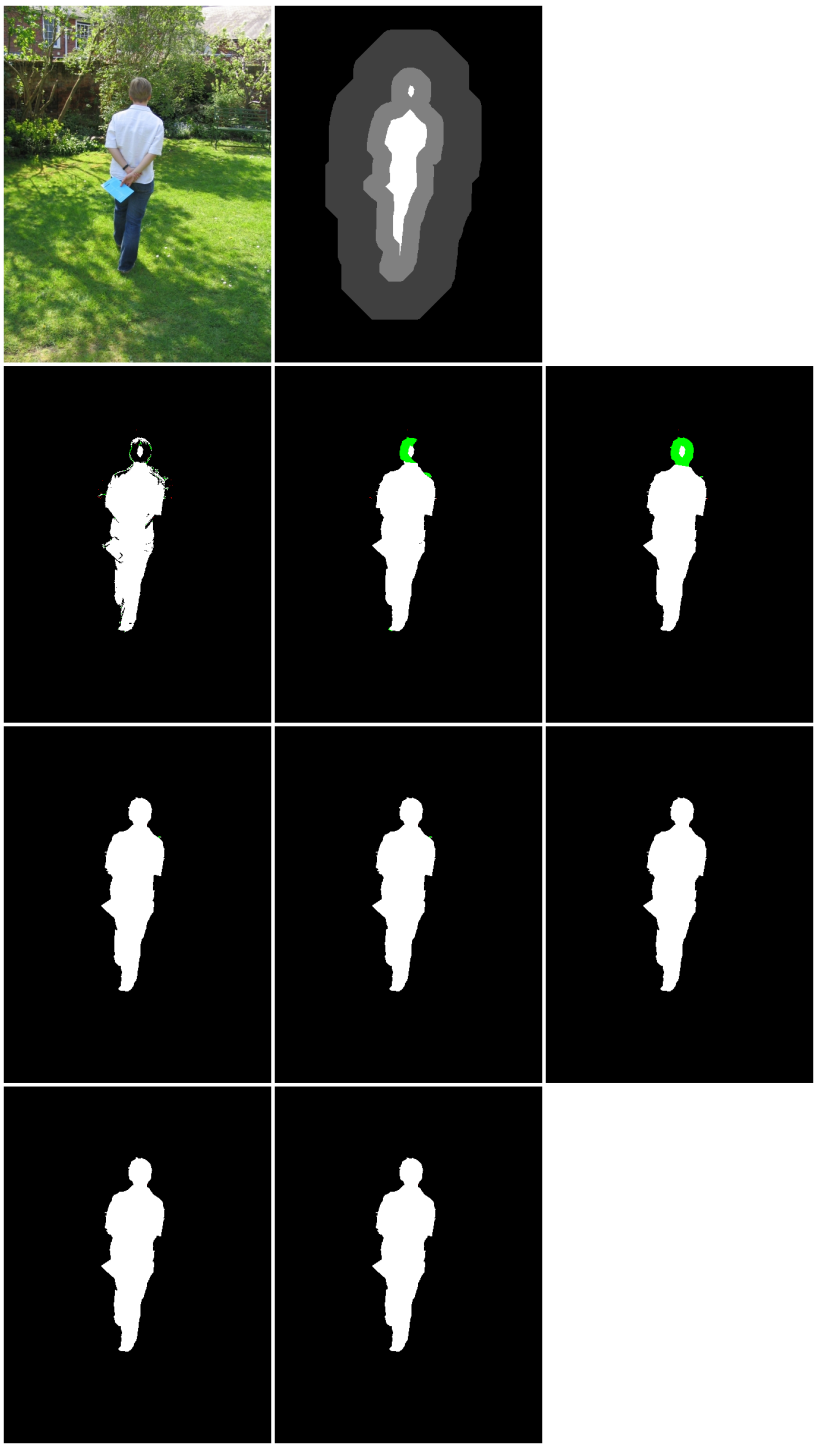


Image ”person6”, average overlap difference 1.04%

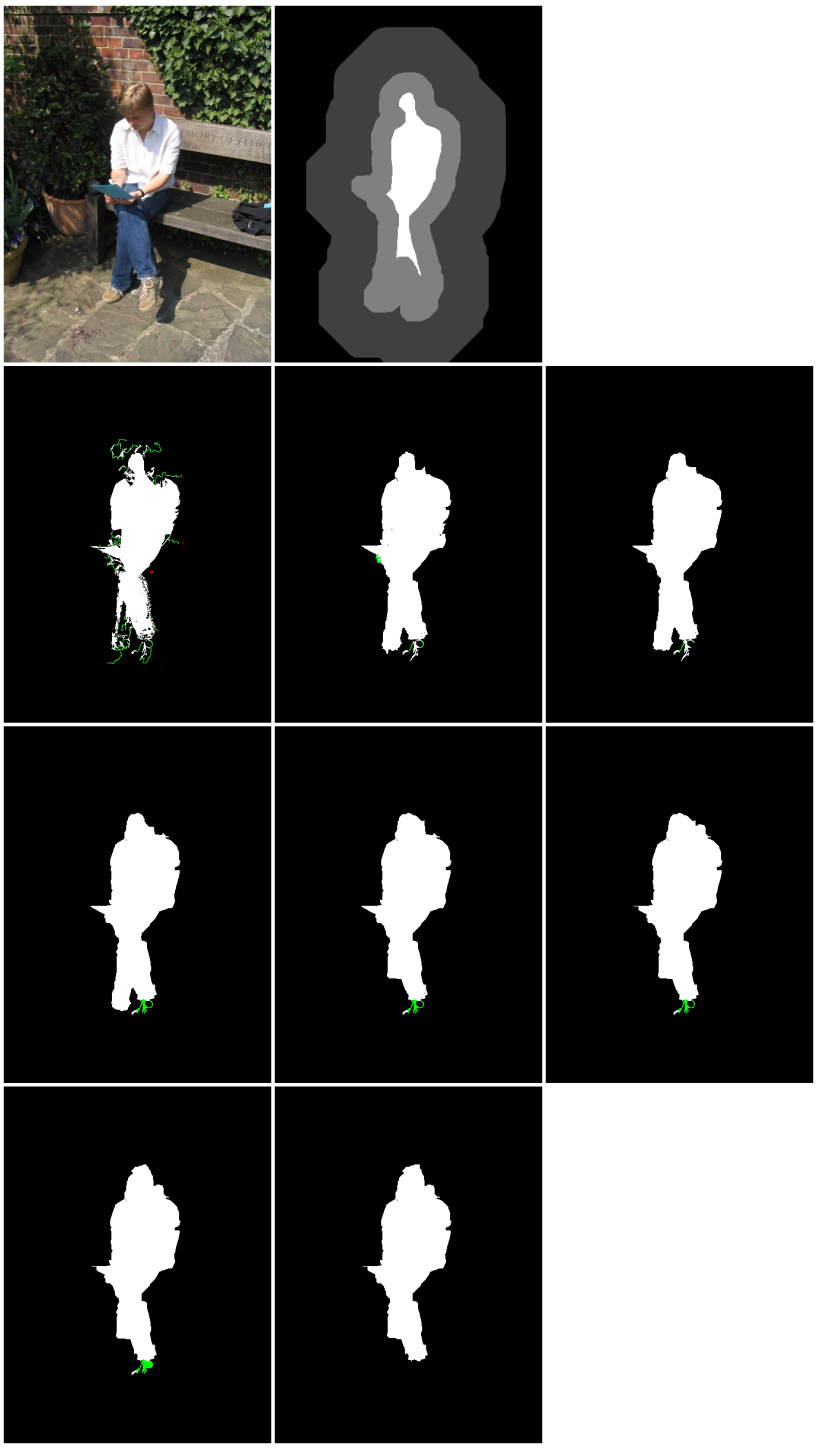


Image ”memorial”, average overlap difference 0.96%

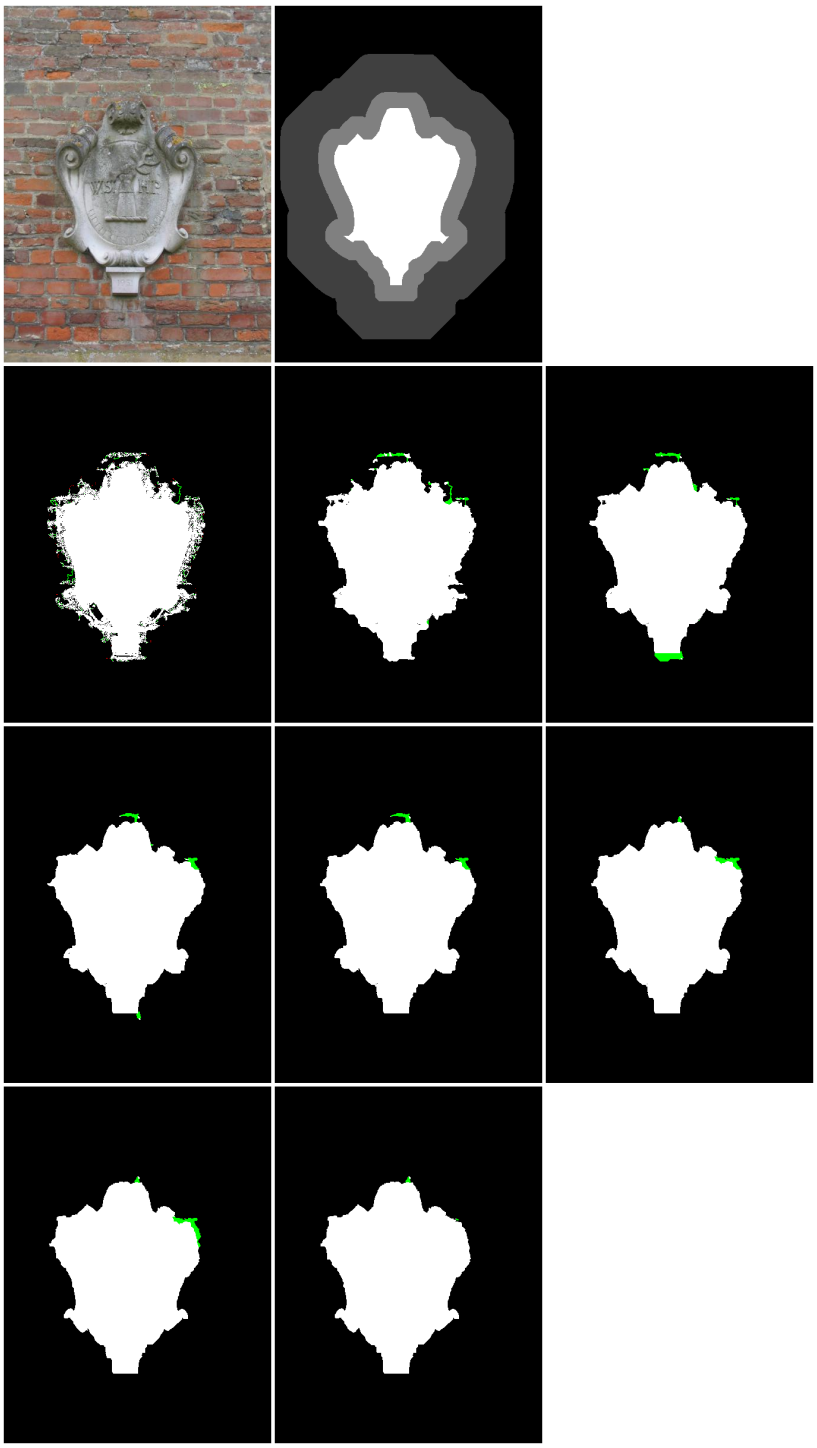


Image ”37073”, average overlap difference 0.86%

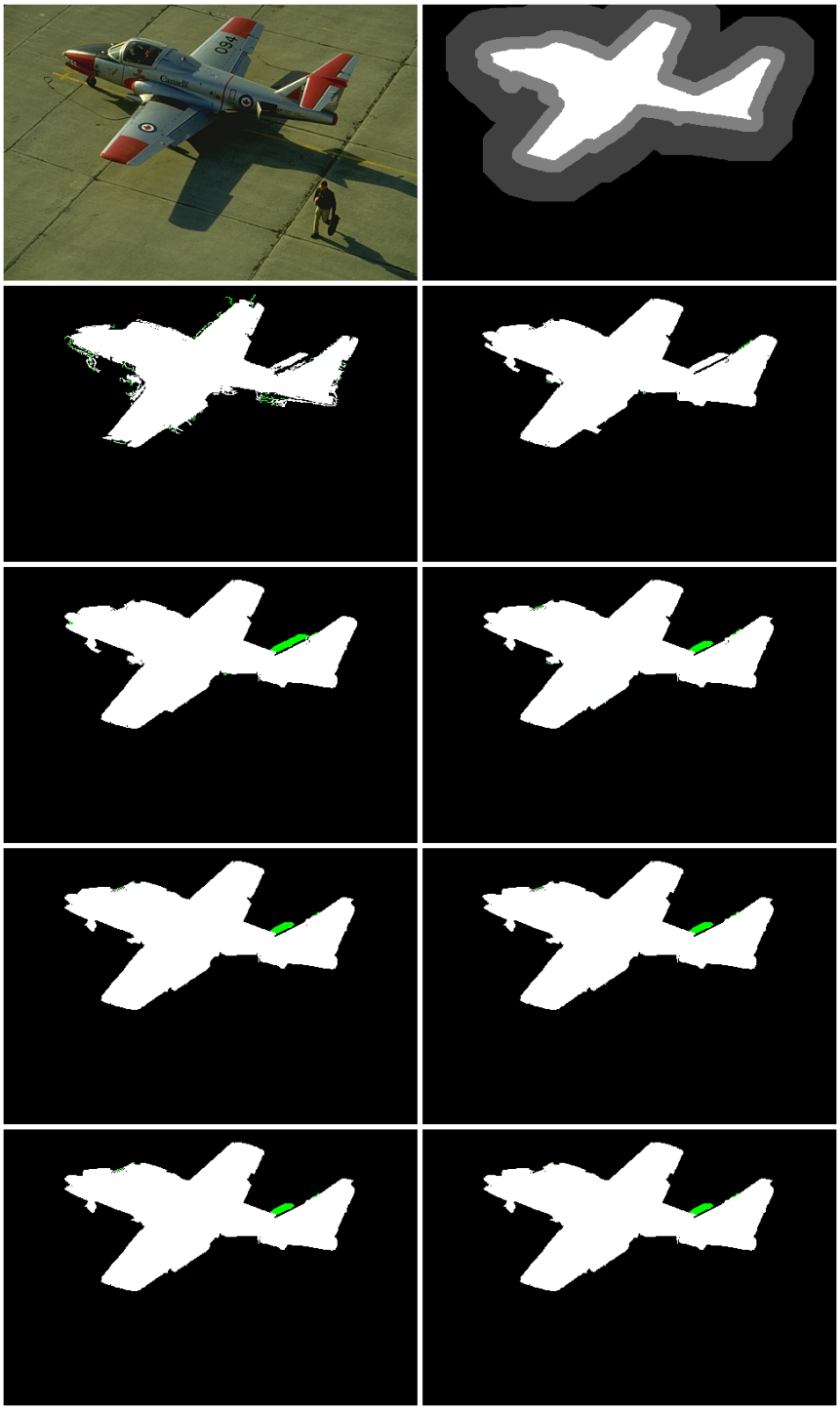


Image ”271008”, average overlap difference 0.69%

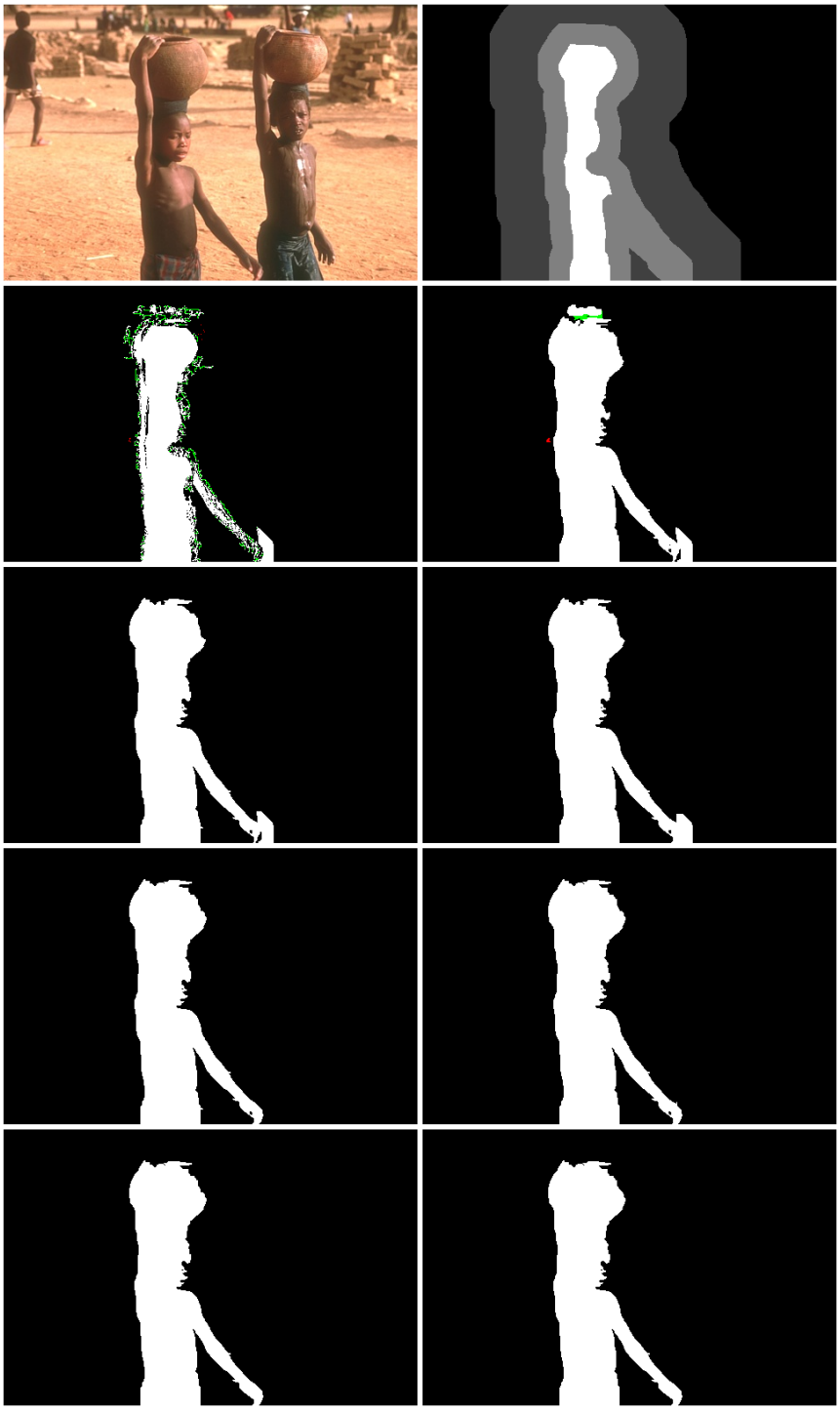


Image ”181079”, average overlap difference 0.47%

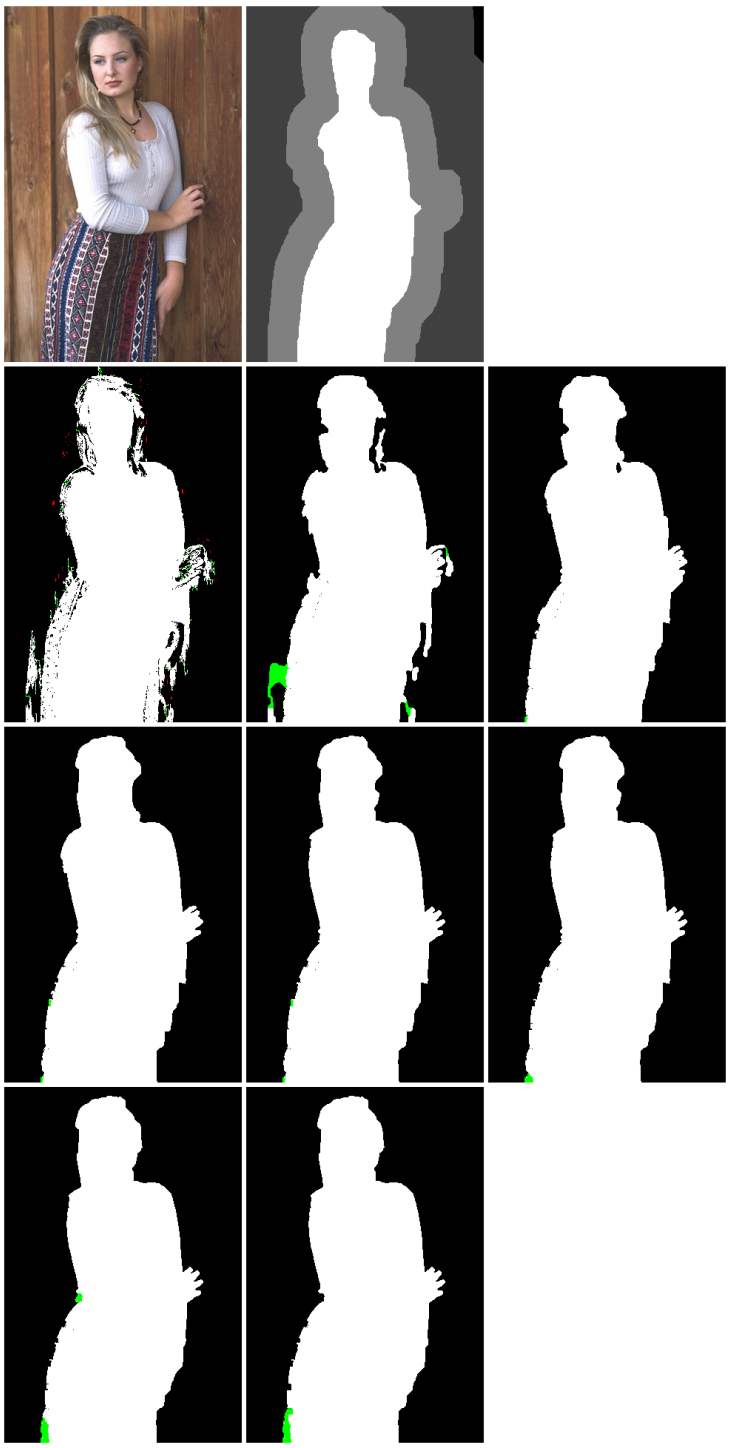


Image ”153093”, average overlap difference 0.43%

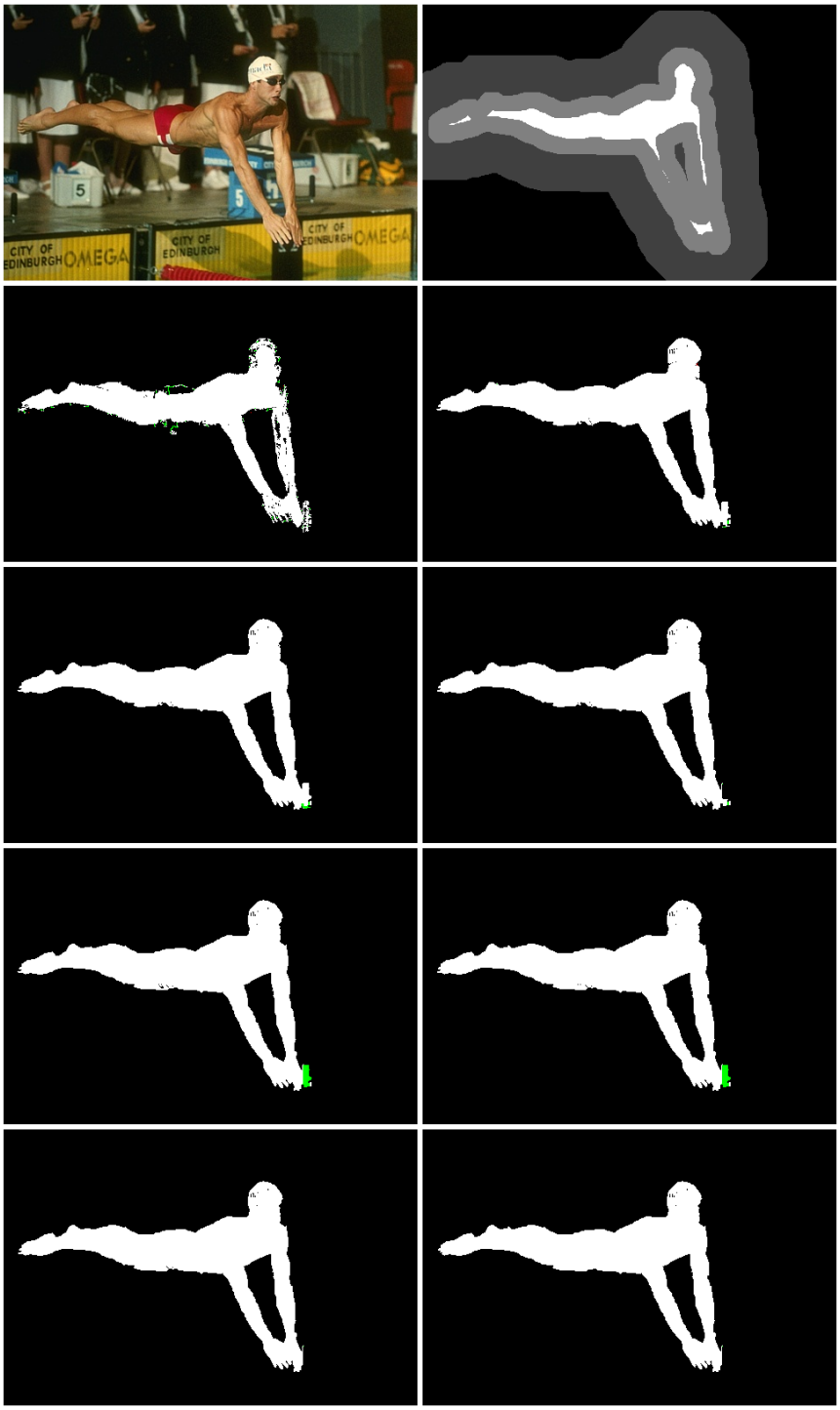


Image ”209070”, average overlap difference 0.42%

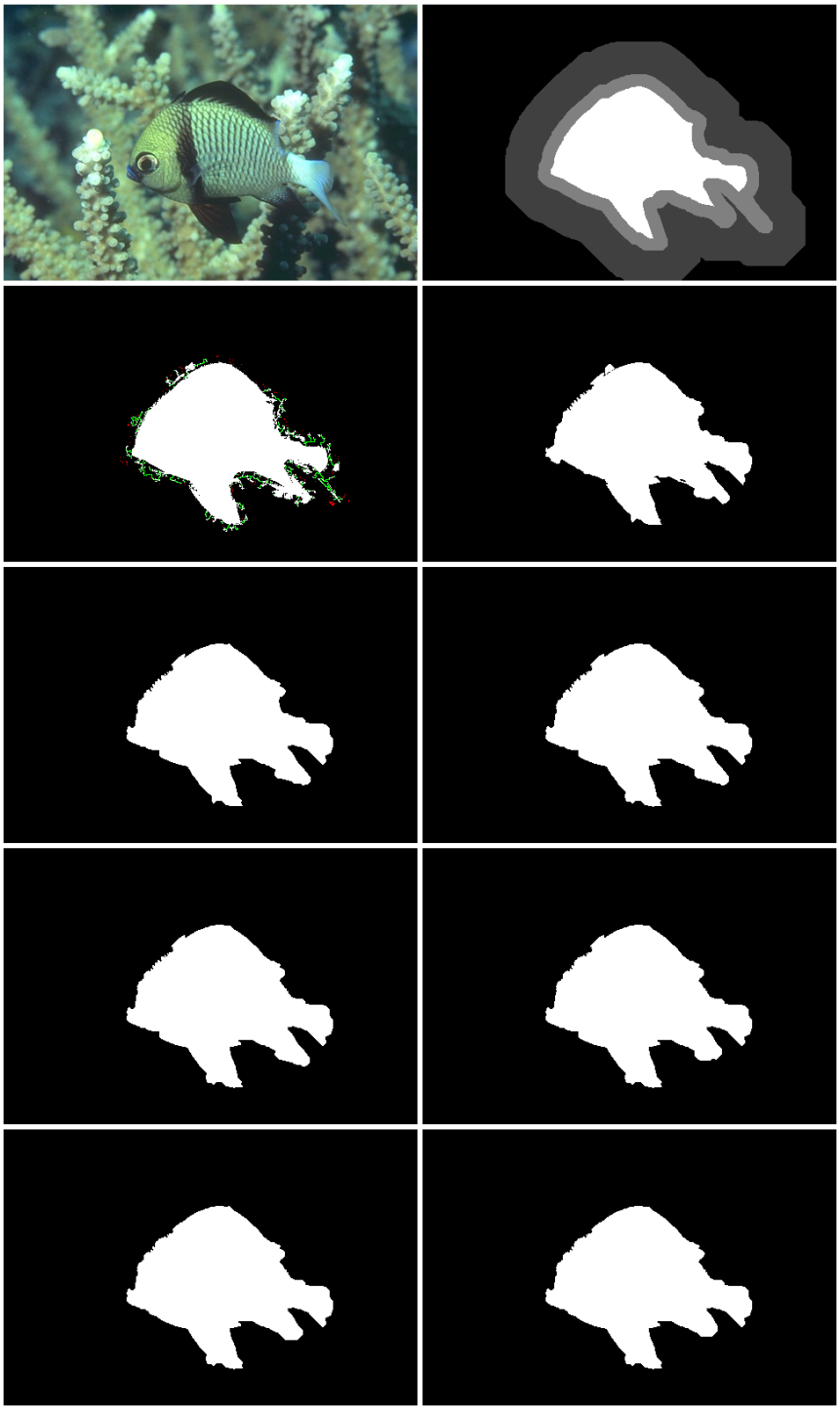


Image ”69020”, average overlap difference 0.38%

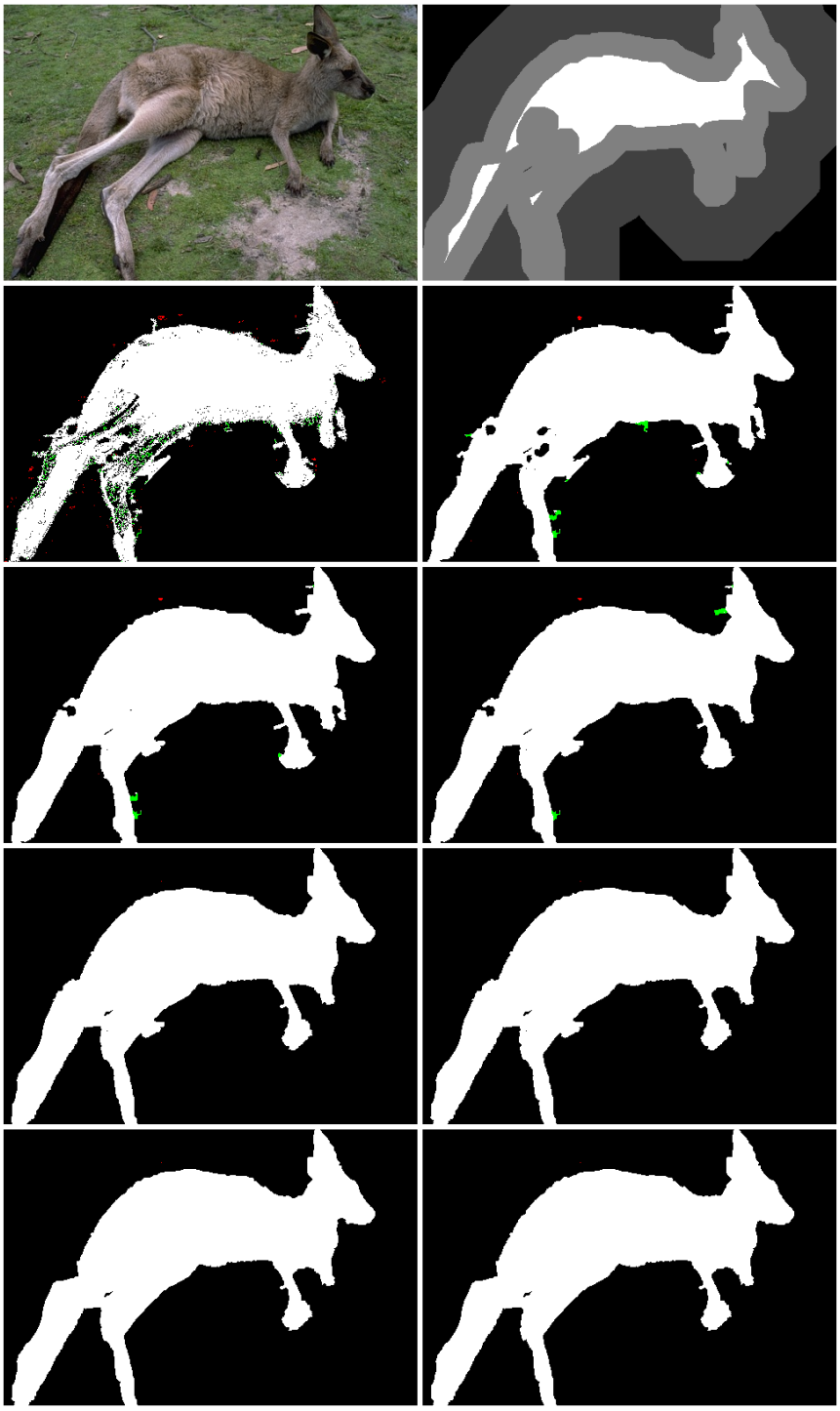


Image ”llama”, average overlap difference 0.38%

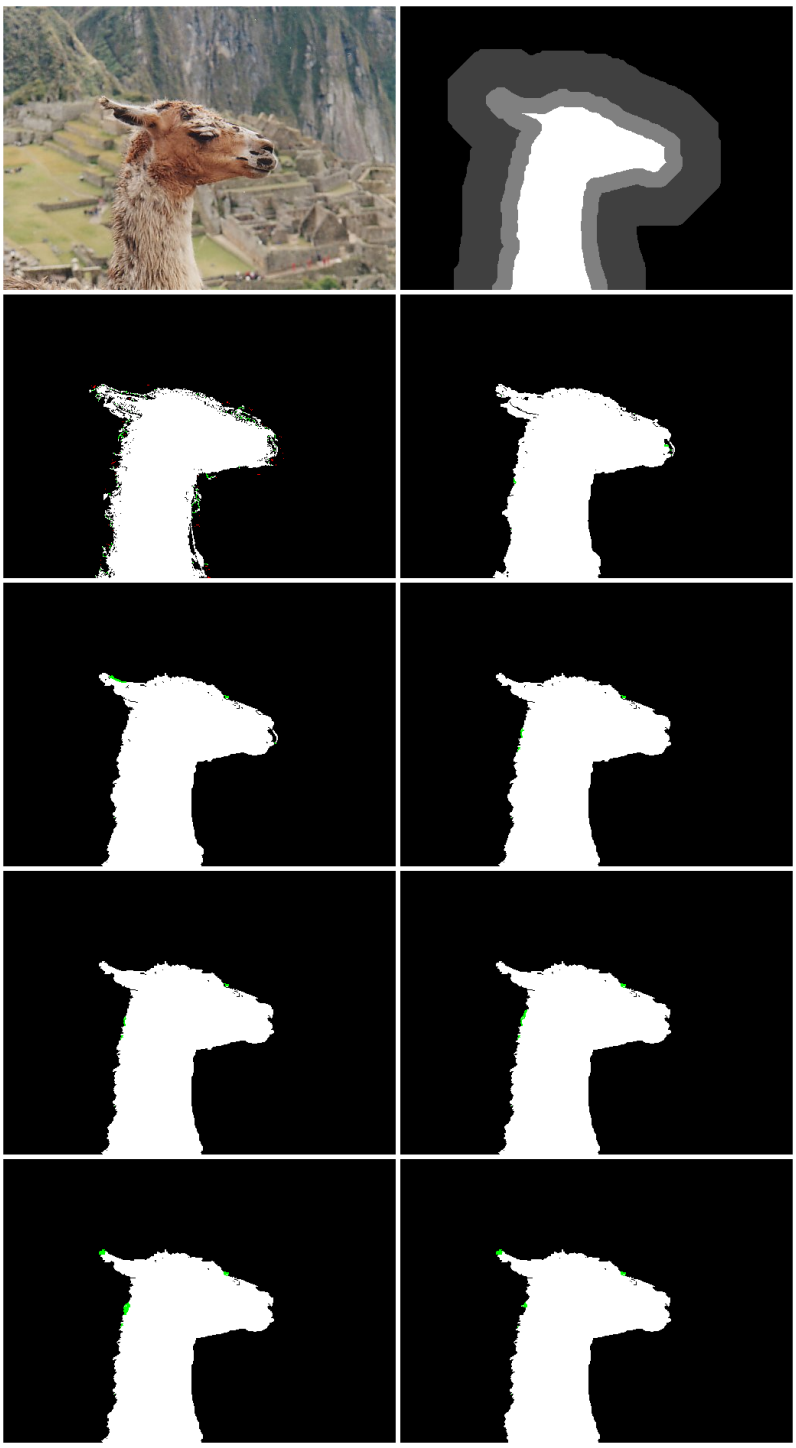


Image ”banana3”, average overlap difference 0.33%

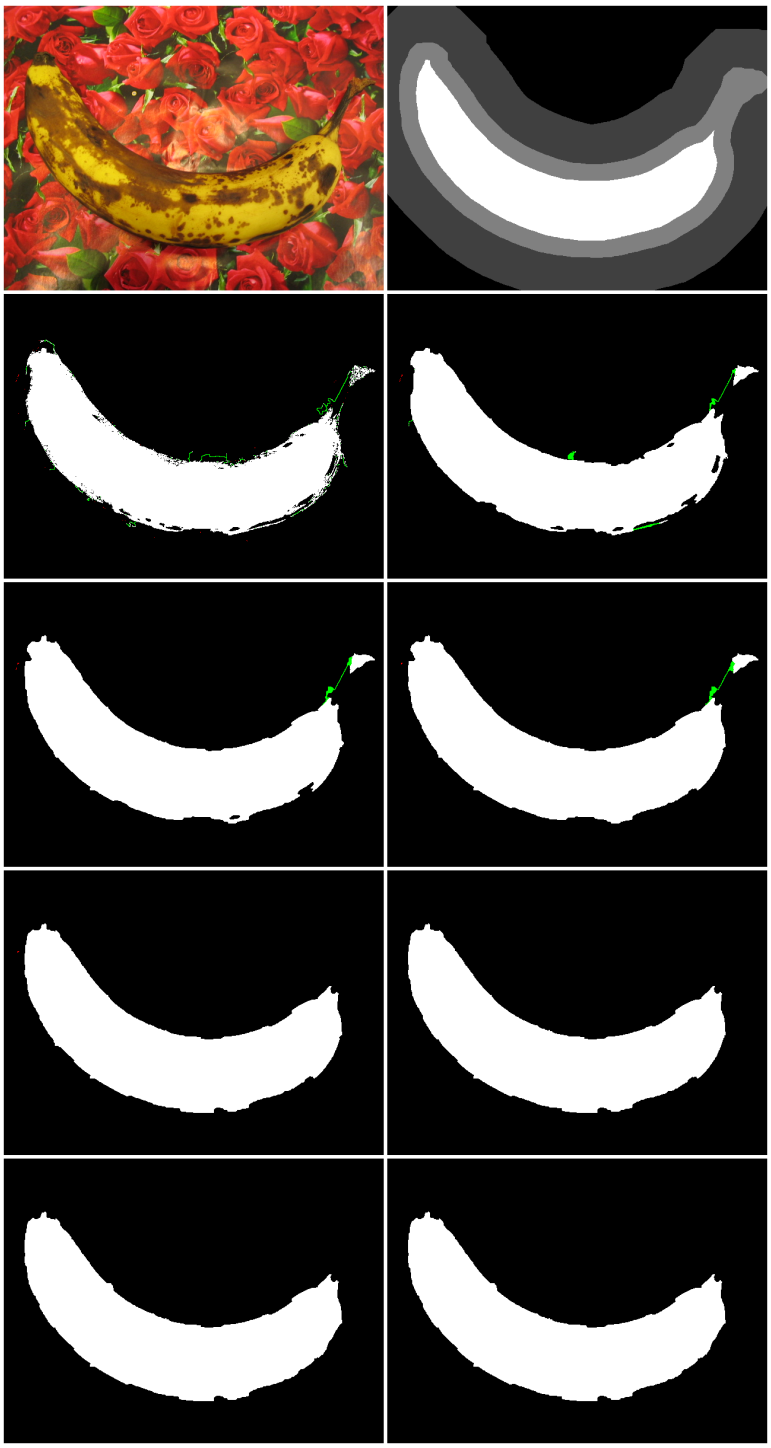


Image ”banana2”, average overlap difference 0.29%

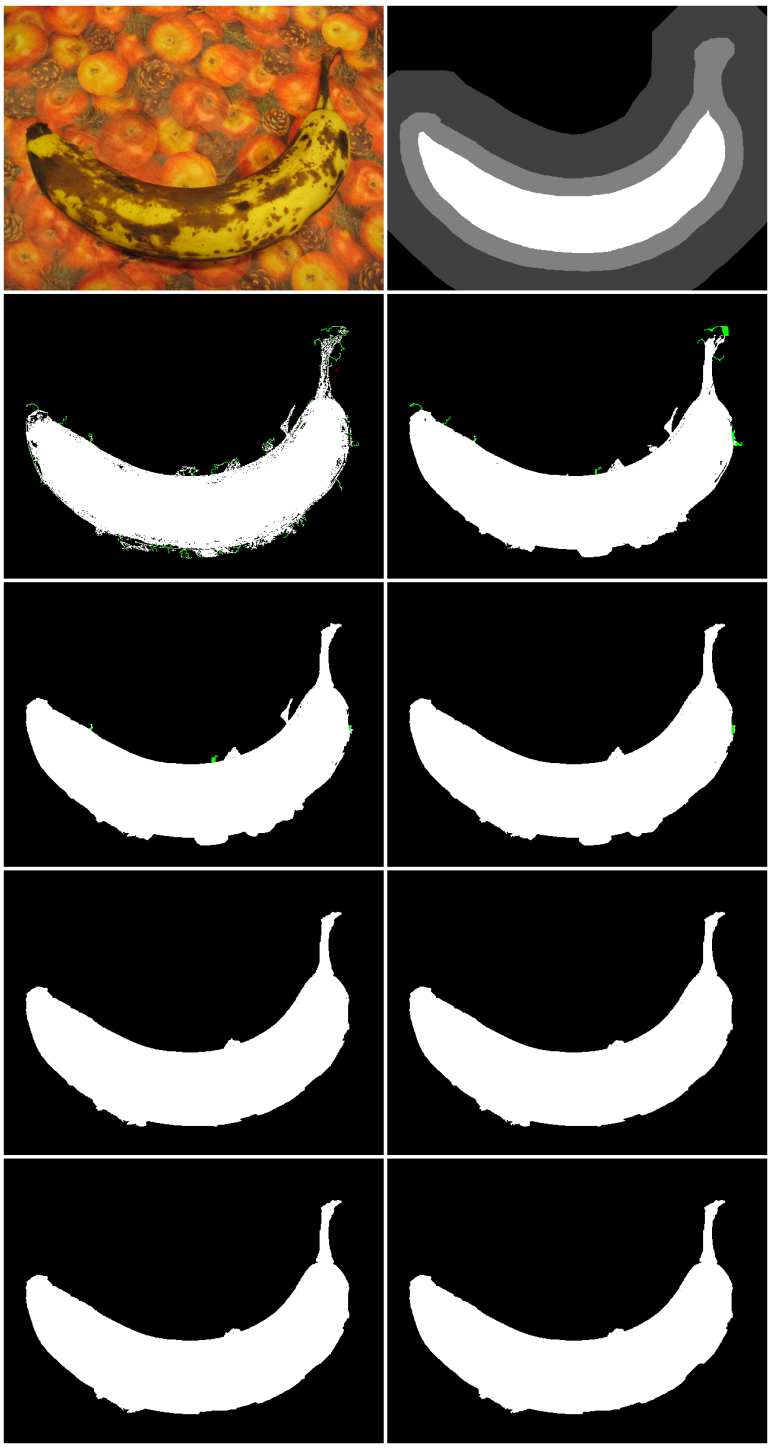


Image ”153077”, average overlap difference 0.28%

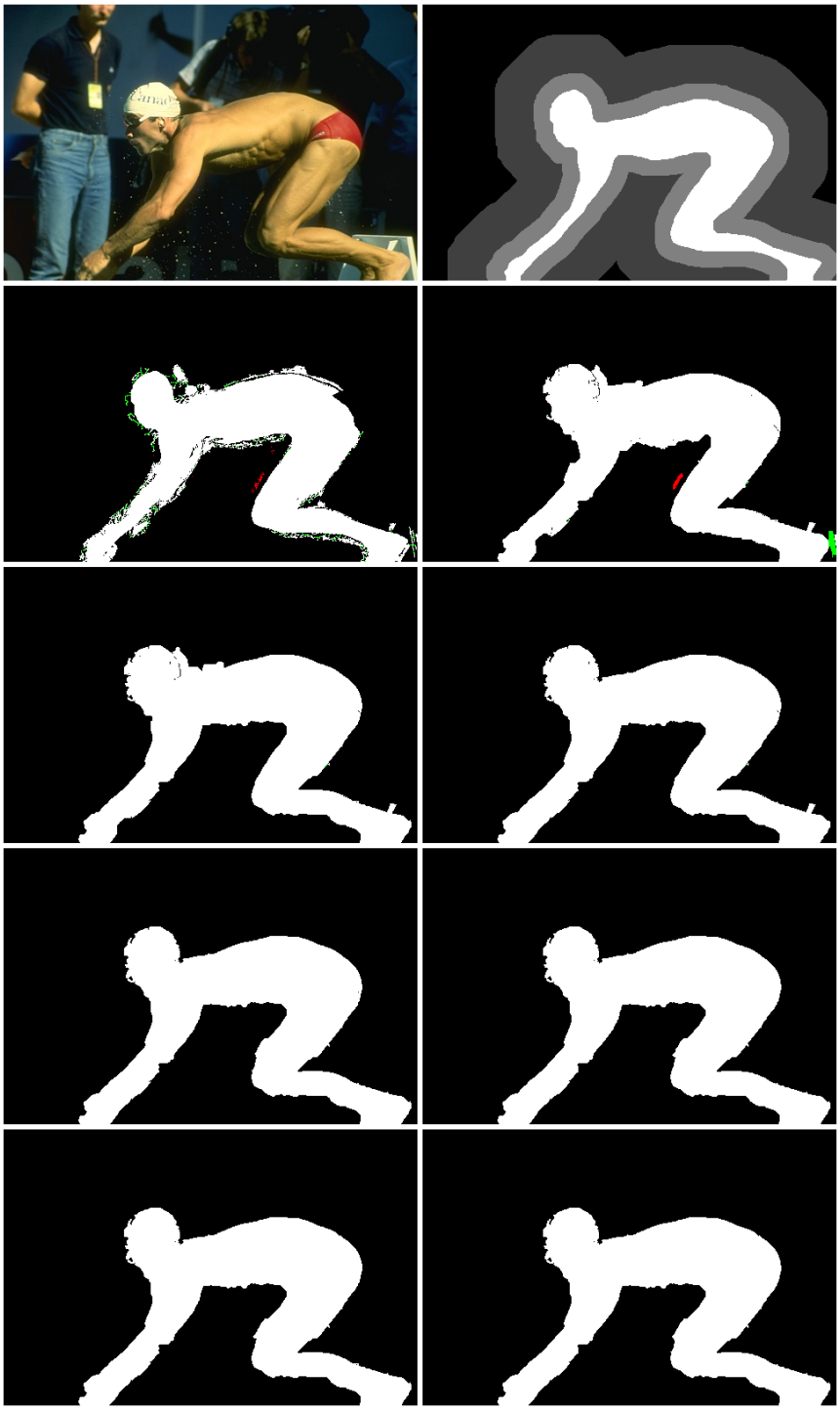


Image ”376043”, average overlap difference 0.27%

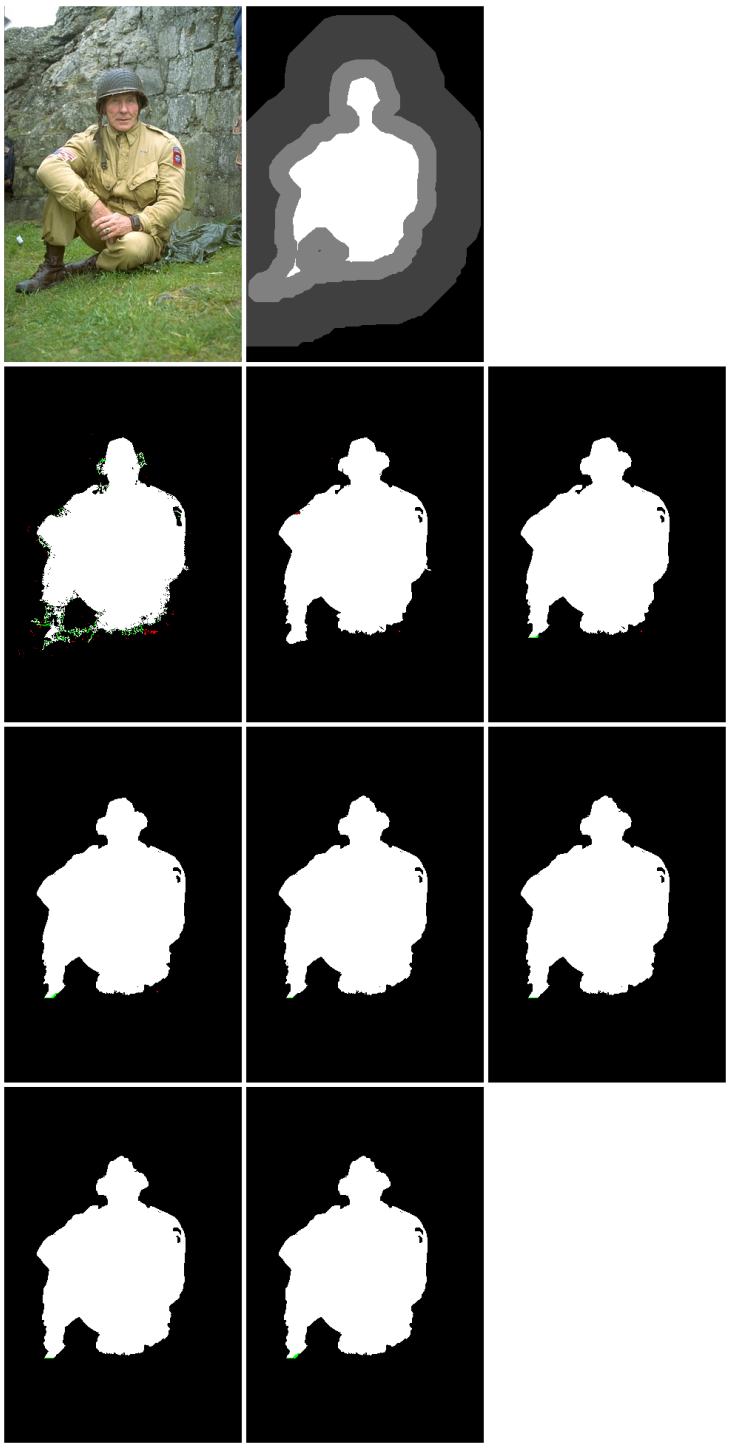


Image ”doll”, average overlap difference 0.24%

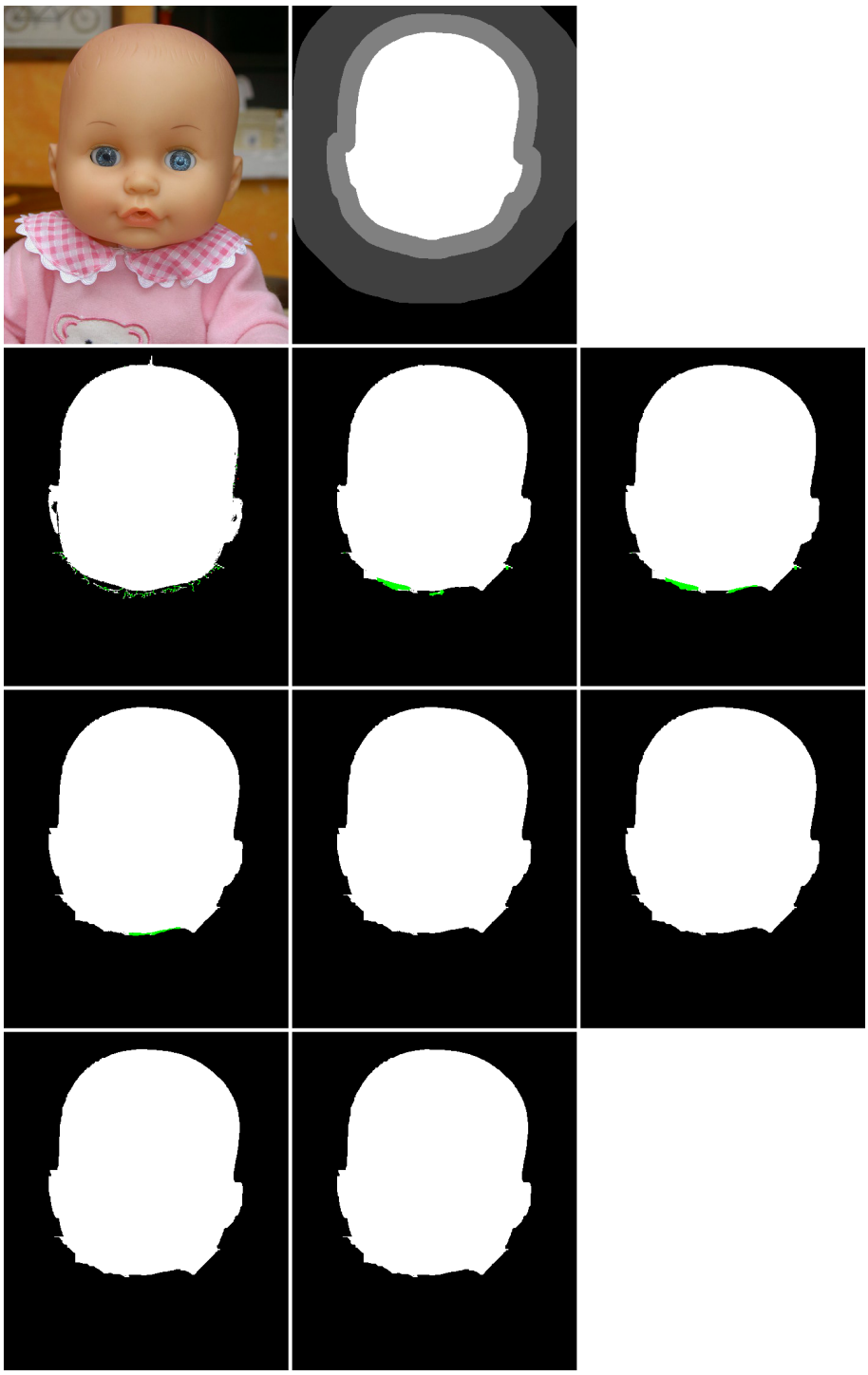


Image ”21077”, average overlap difference 0.23%

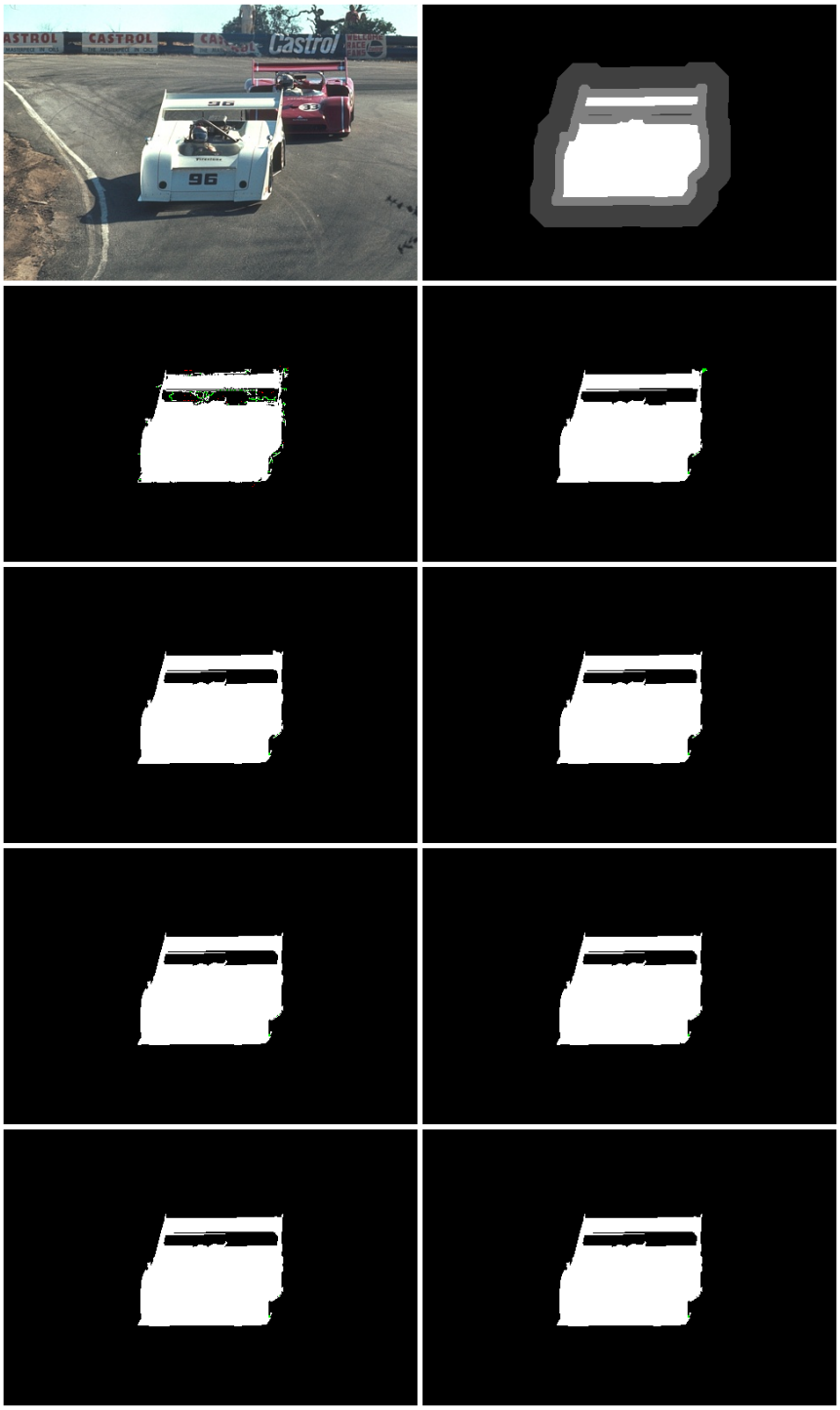


Image ”person8”, average overlap difference 0.21%

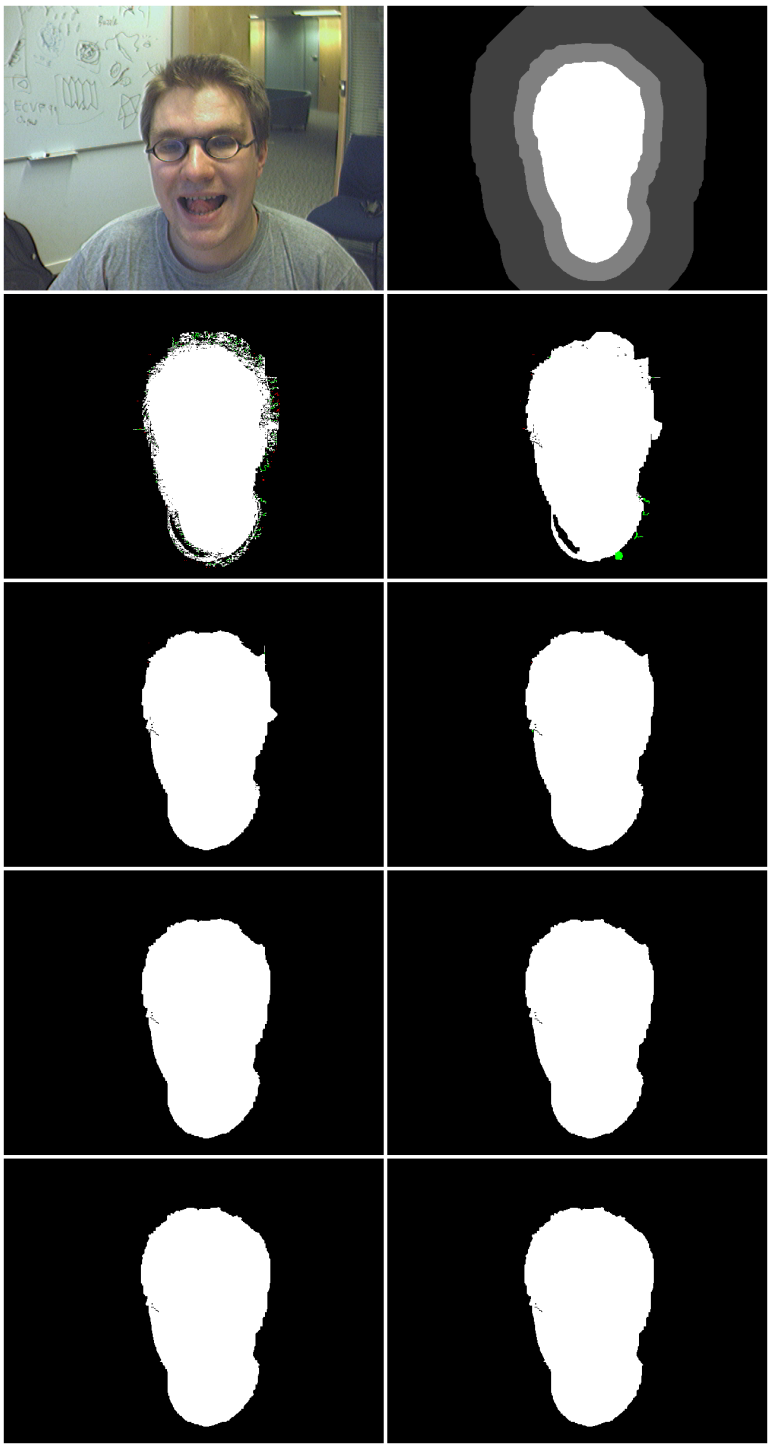


Image ”208001”, average overlap difference 0.21%

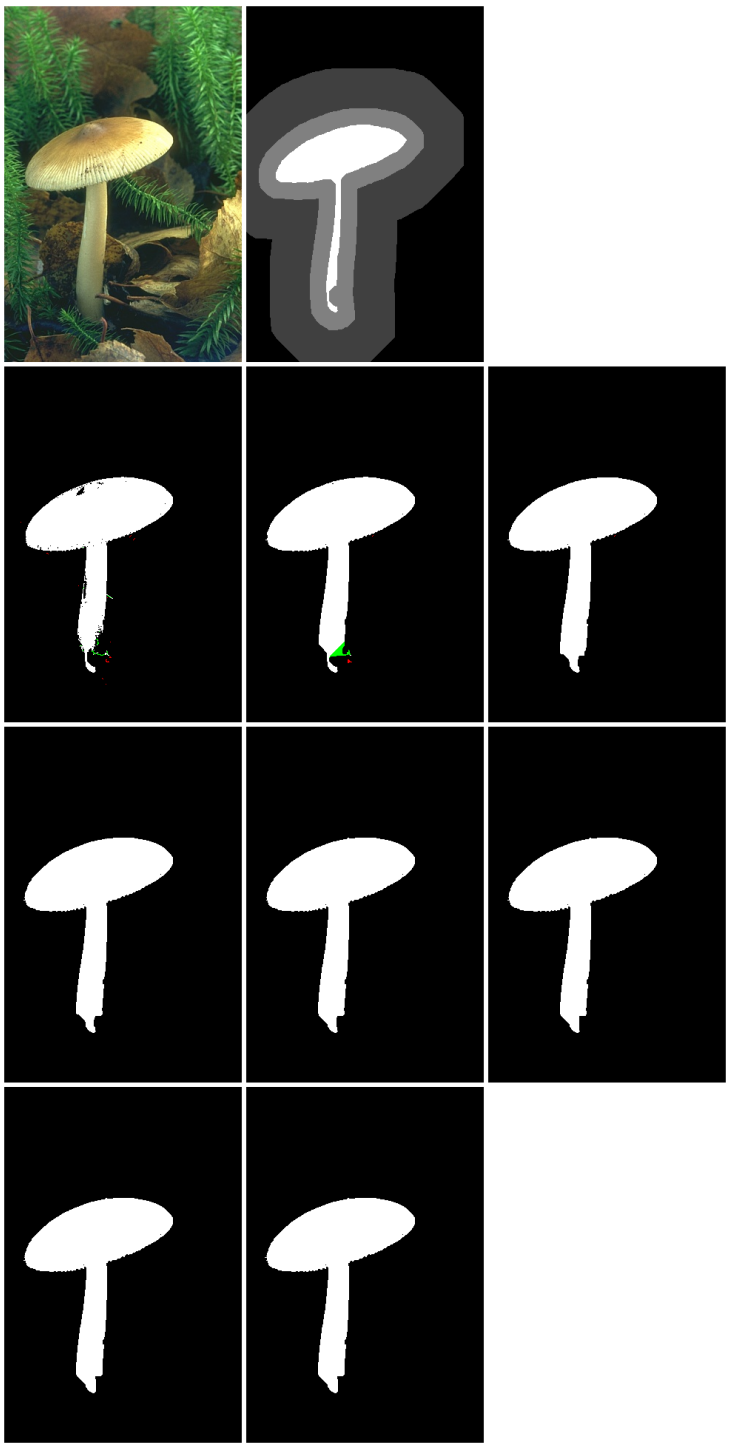


Image ”65019”, average overlap difference 0.20%

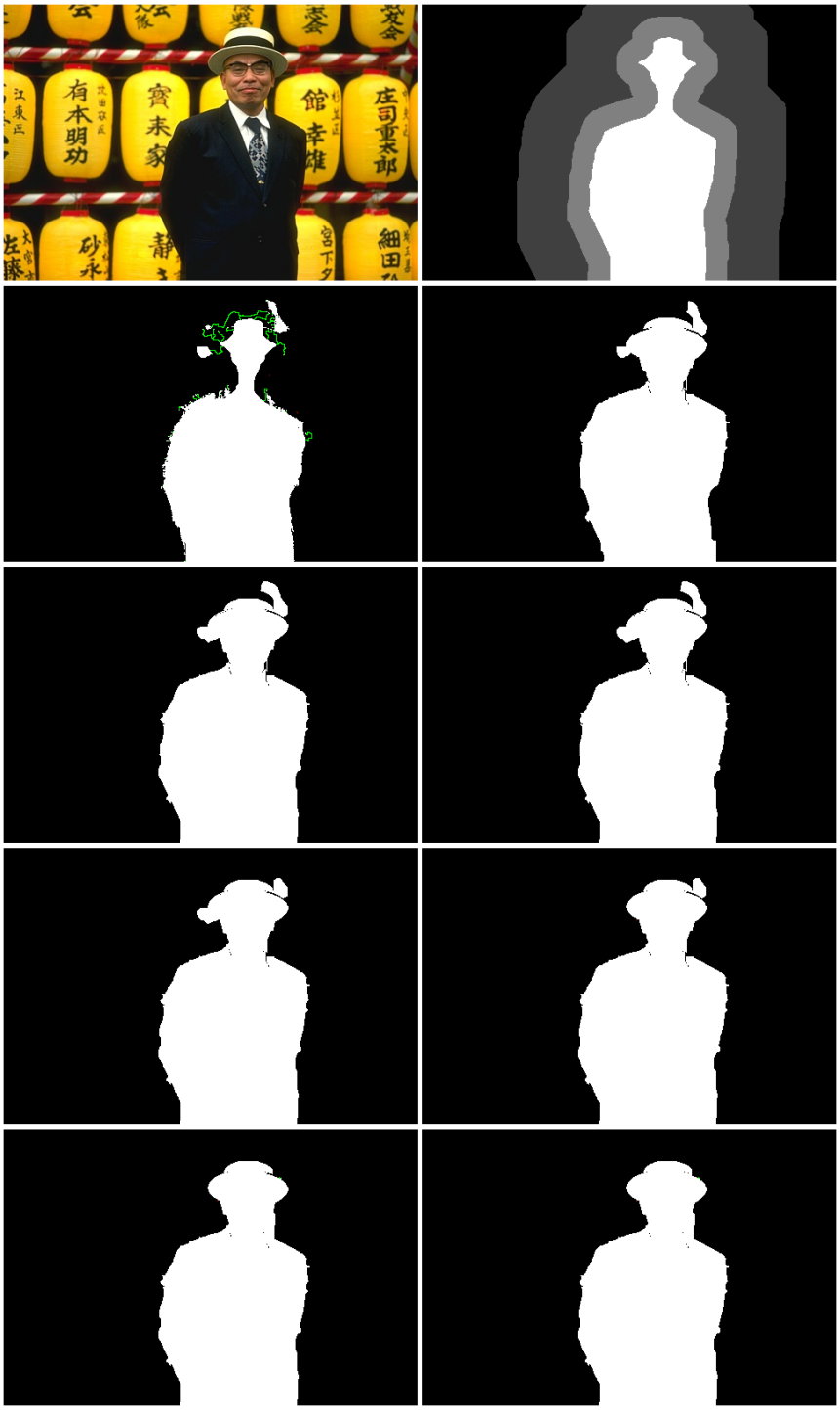


Image ”flower”, average overlap difference 0.18%

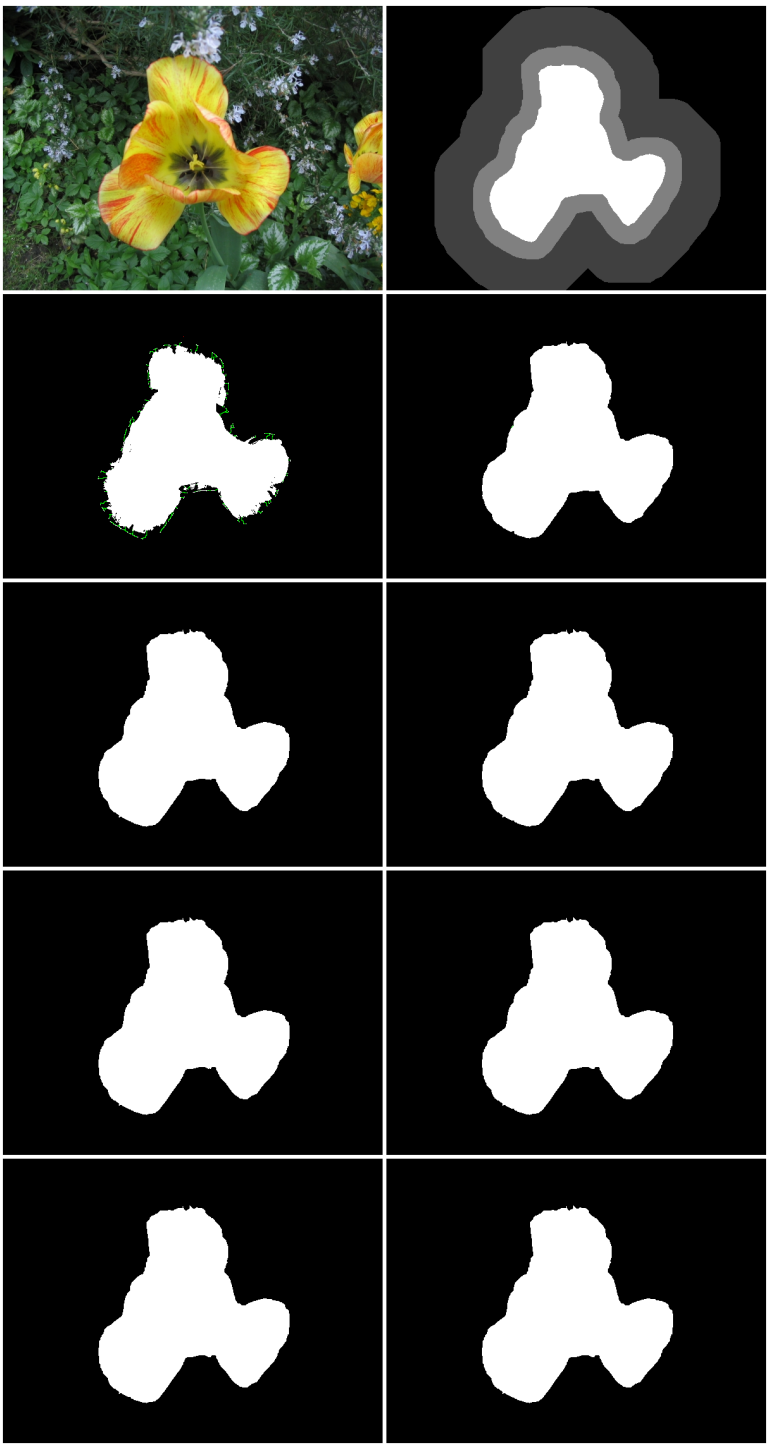


Image ”189080”, average overlap difference 0.16%

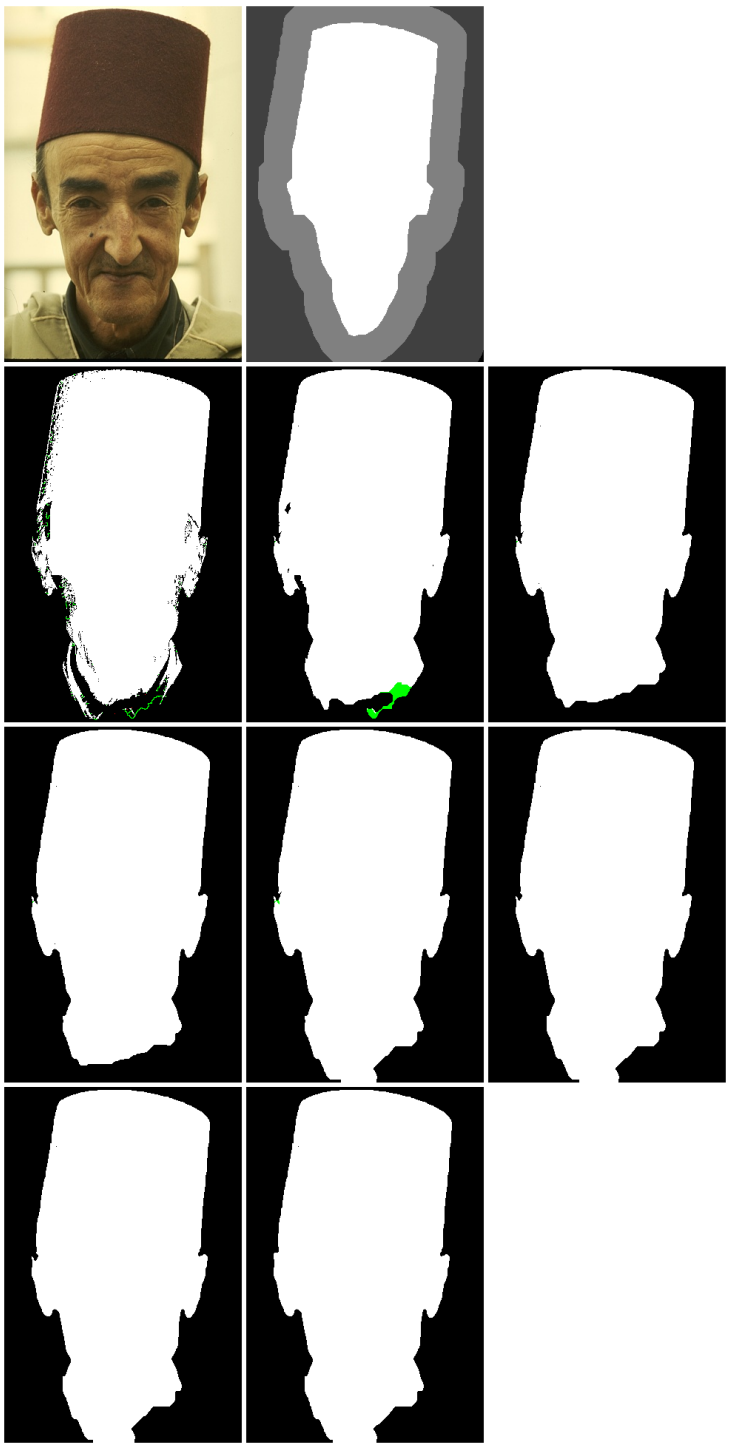


Image ”teddy”, average overlap difference 0.13%

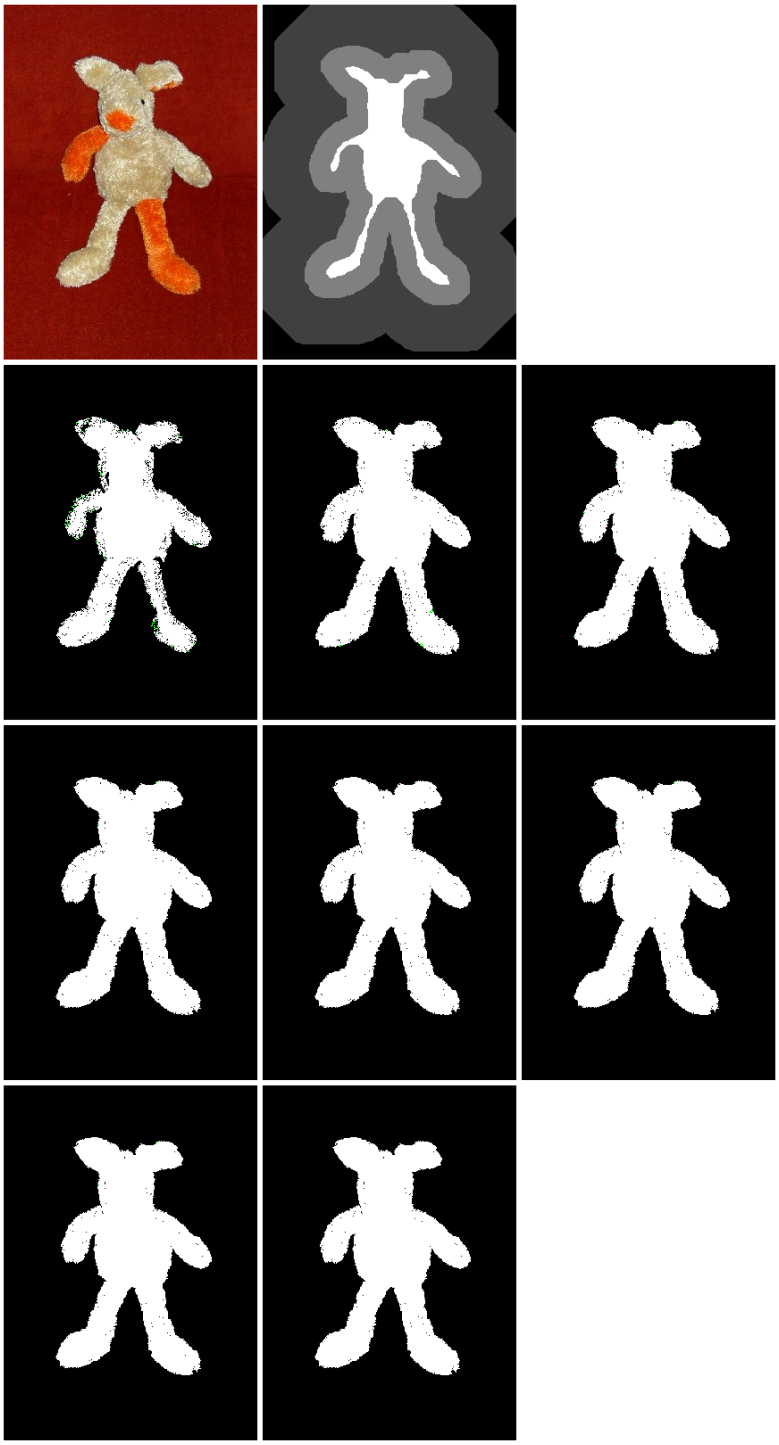


Image ”grave”, average overlap difference 0.12%

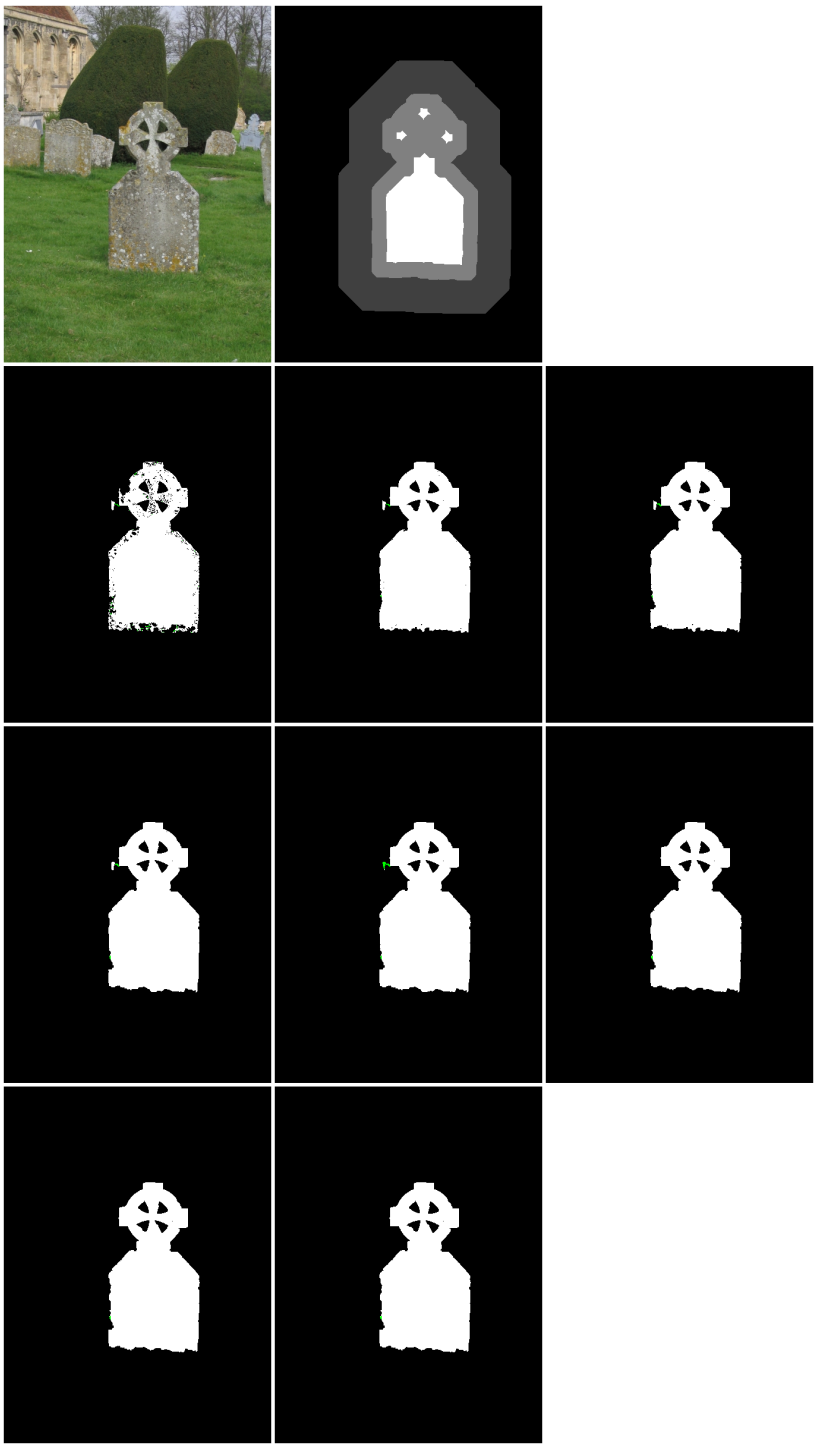


Image ”86016”, average overlap difference 0.09%

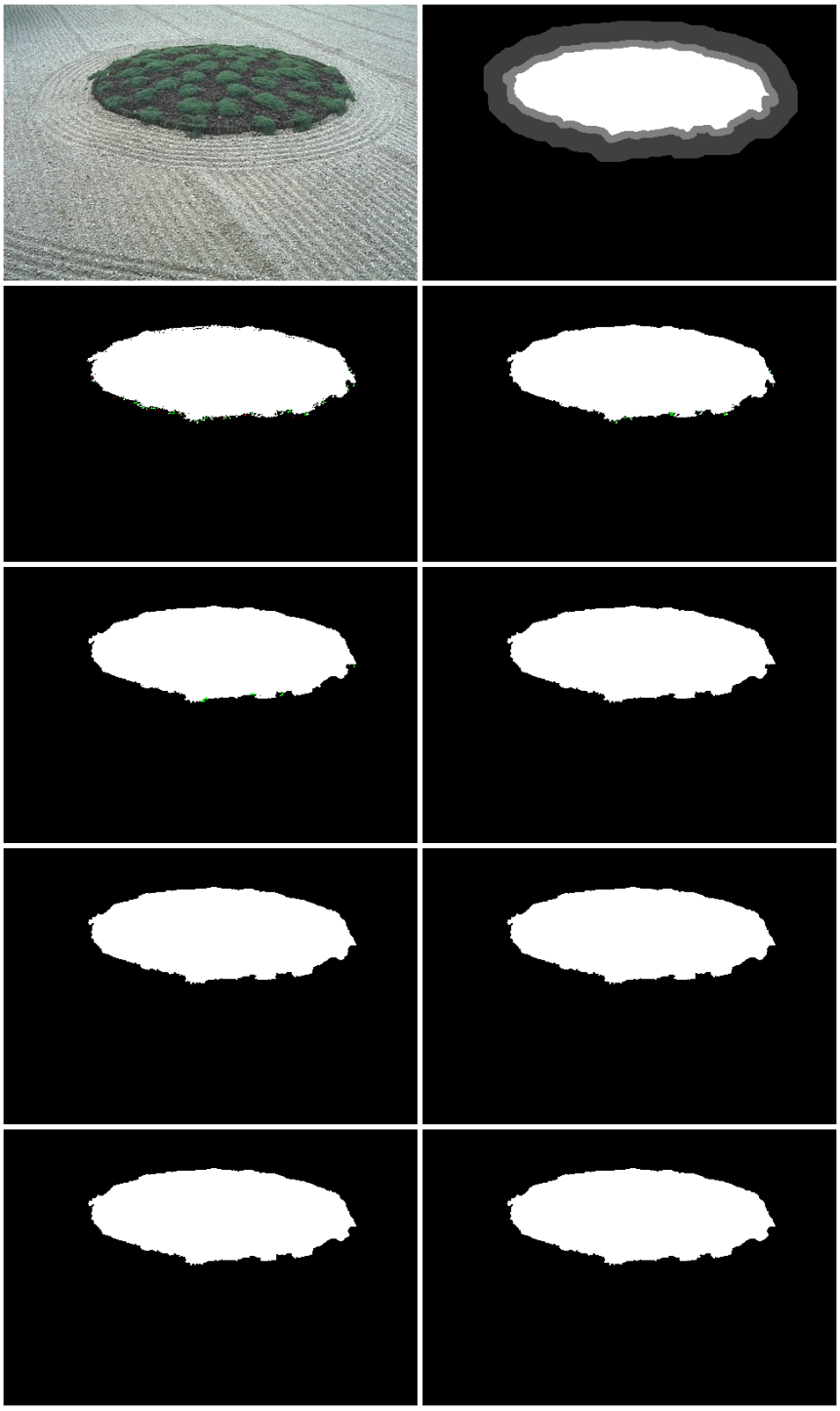


Image ”person1”, average overlap difference 0.09%

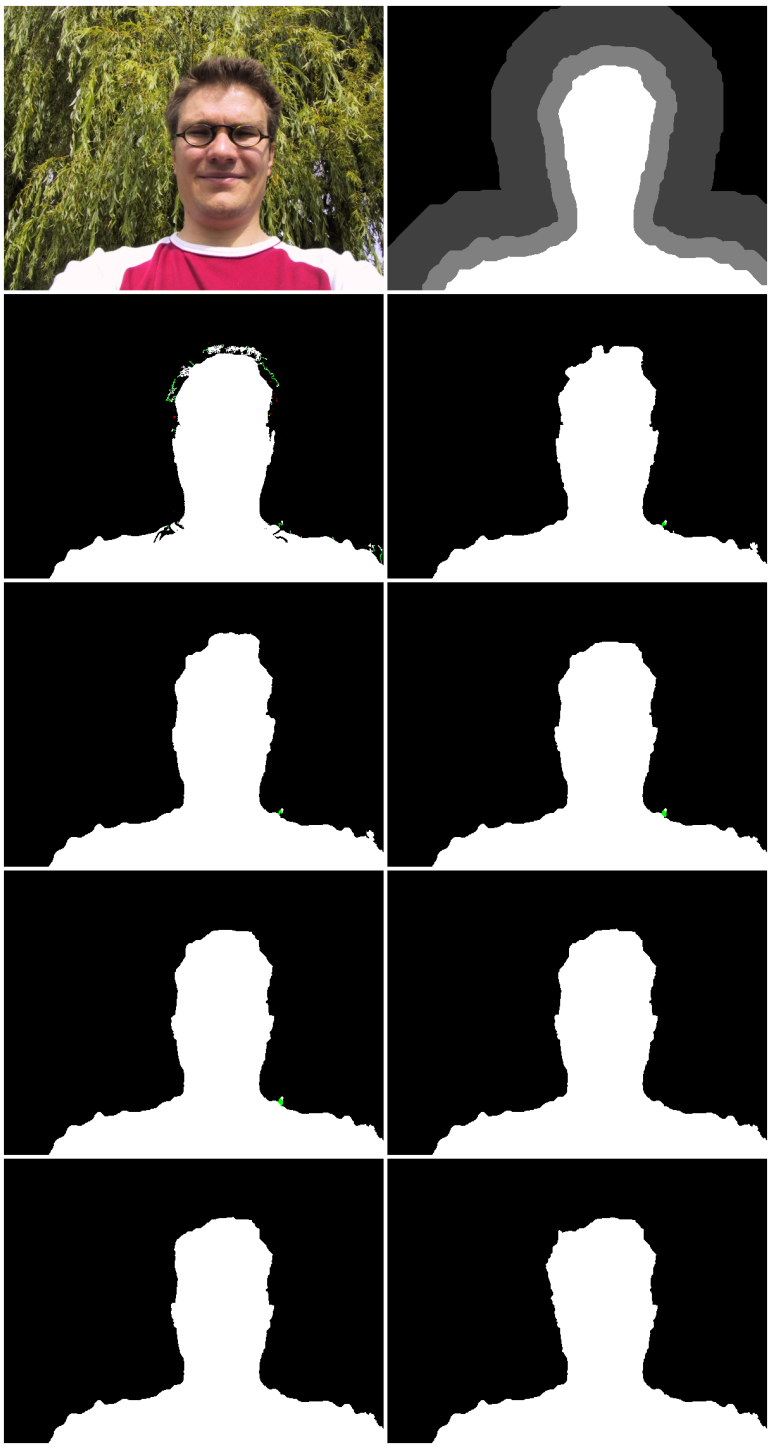


Image ”banana1”, average overlap difference 0.08%

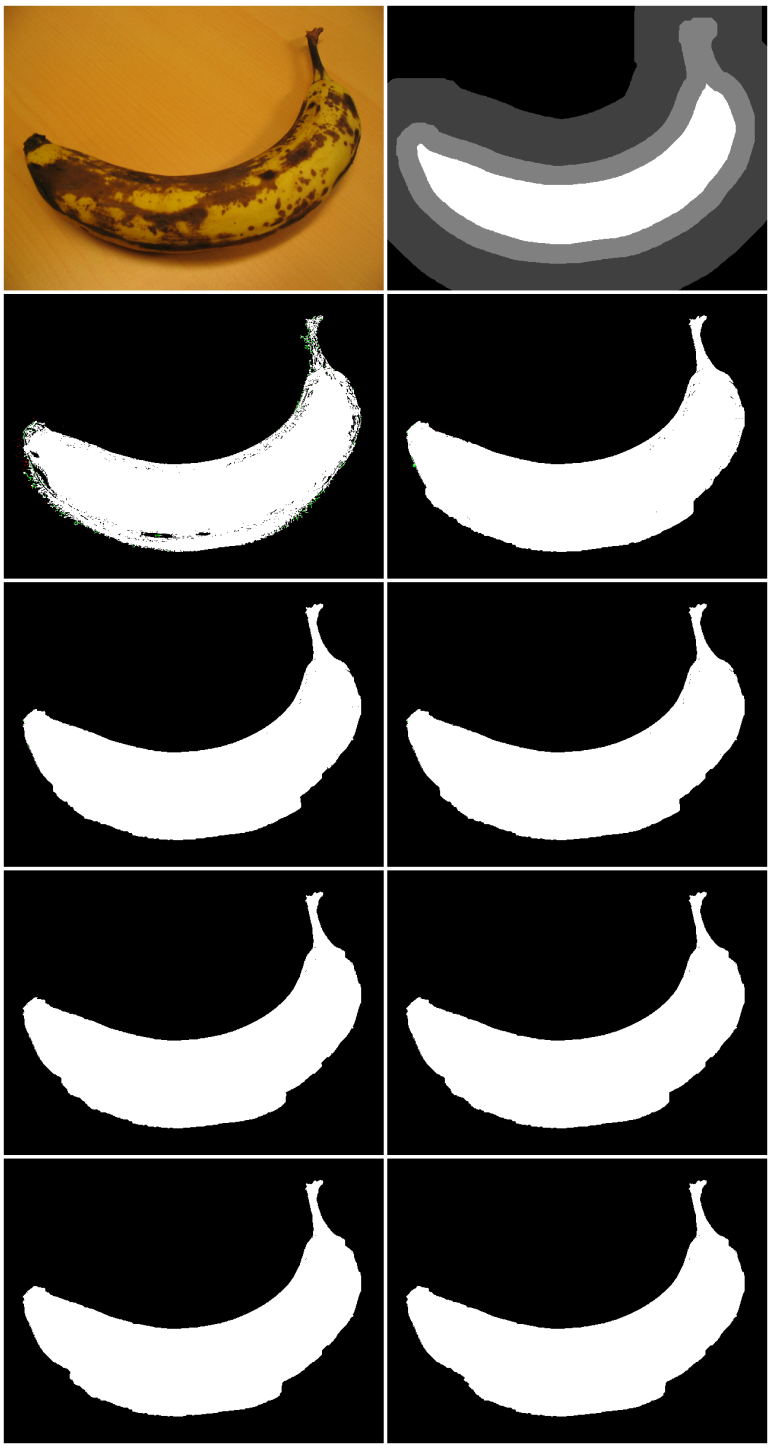


Image ”cross”, average overlap difference 0.08%

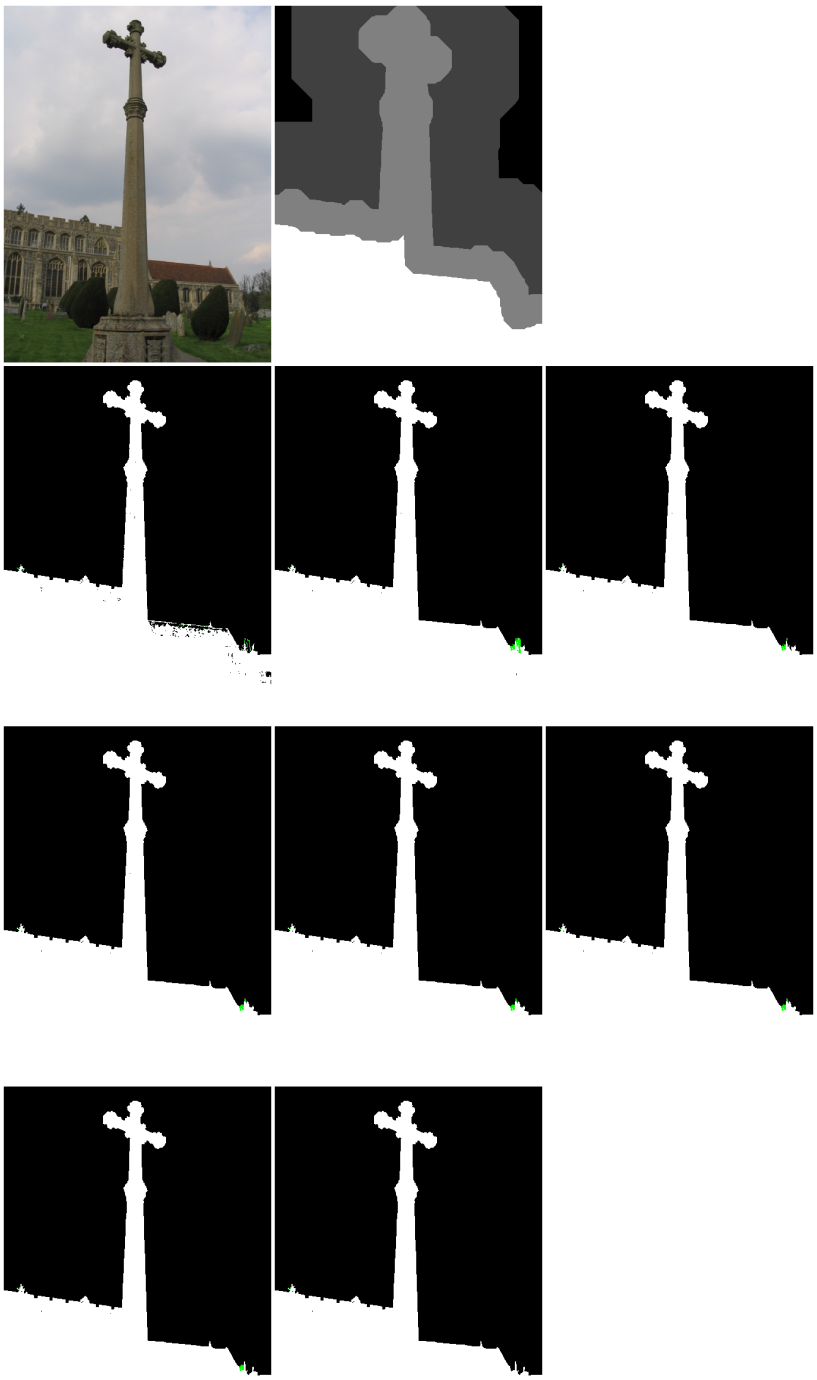


Image ”227092”, average overlap difference 0.08%

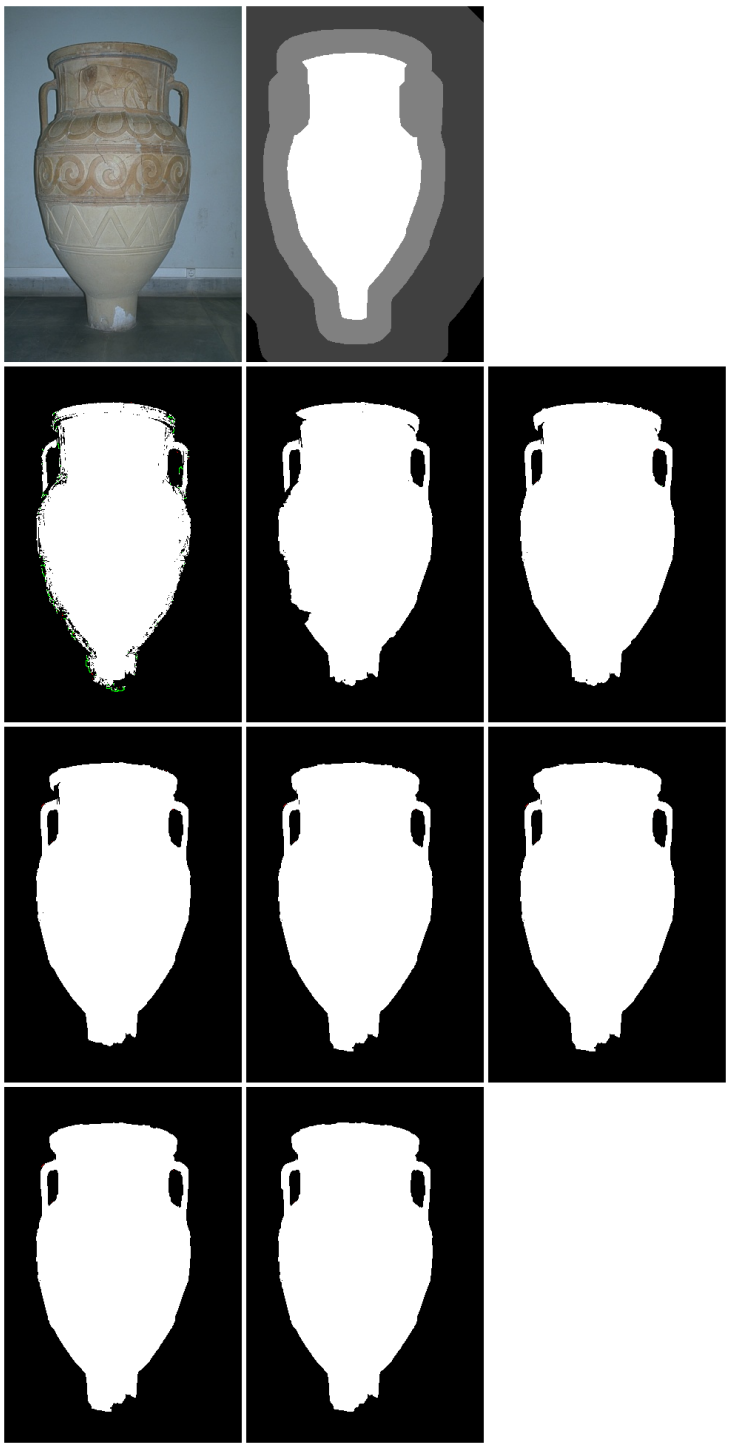


Image ”book”, average overlap difference 0.07%

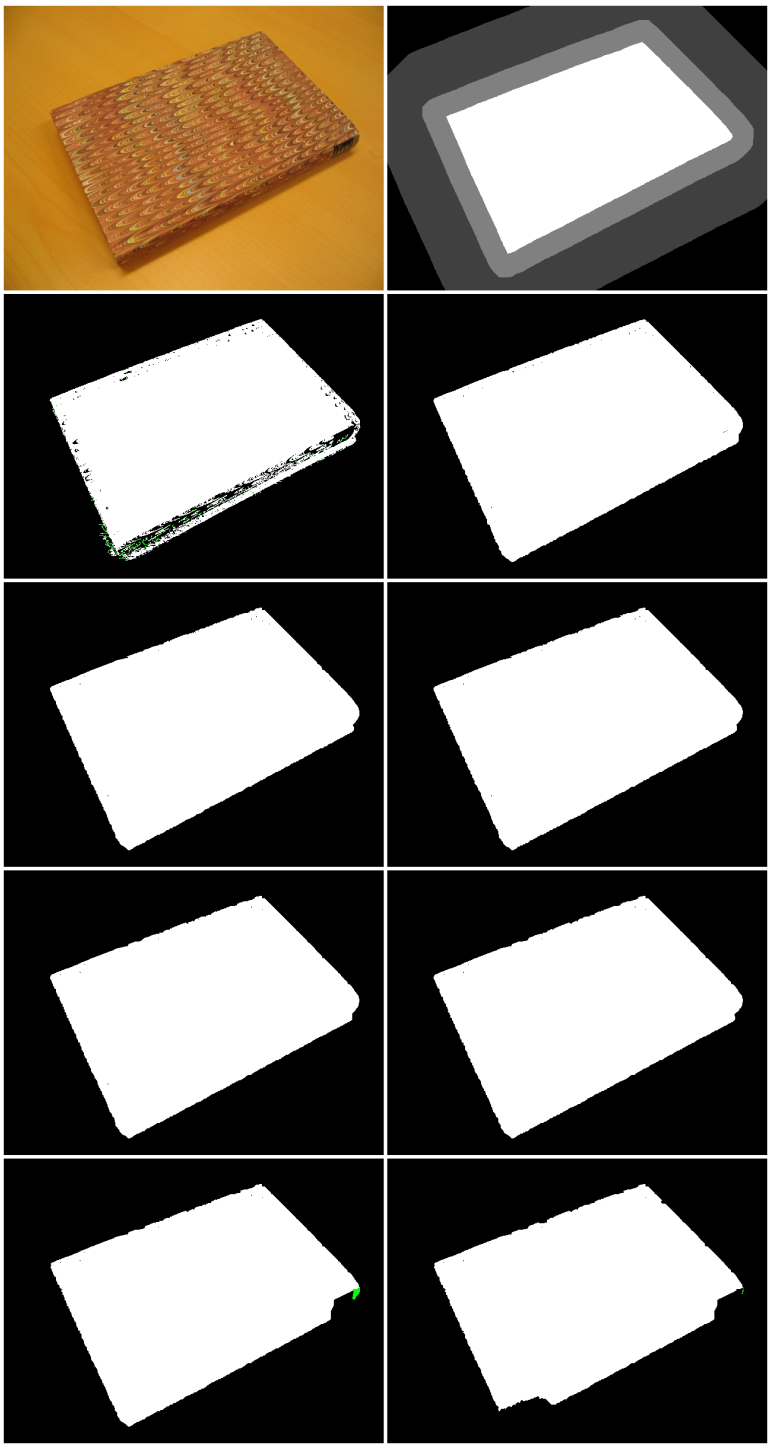


Image ”124080”, average overlap difference 0.06%

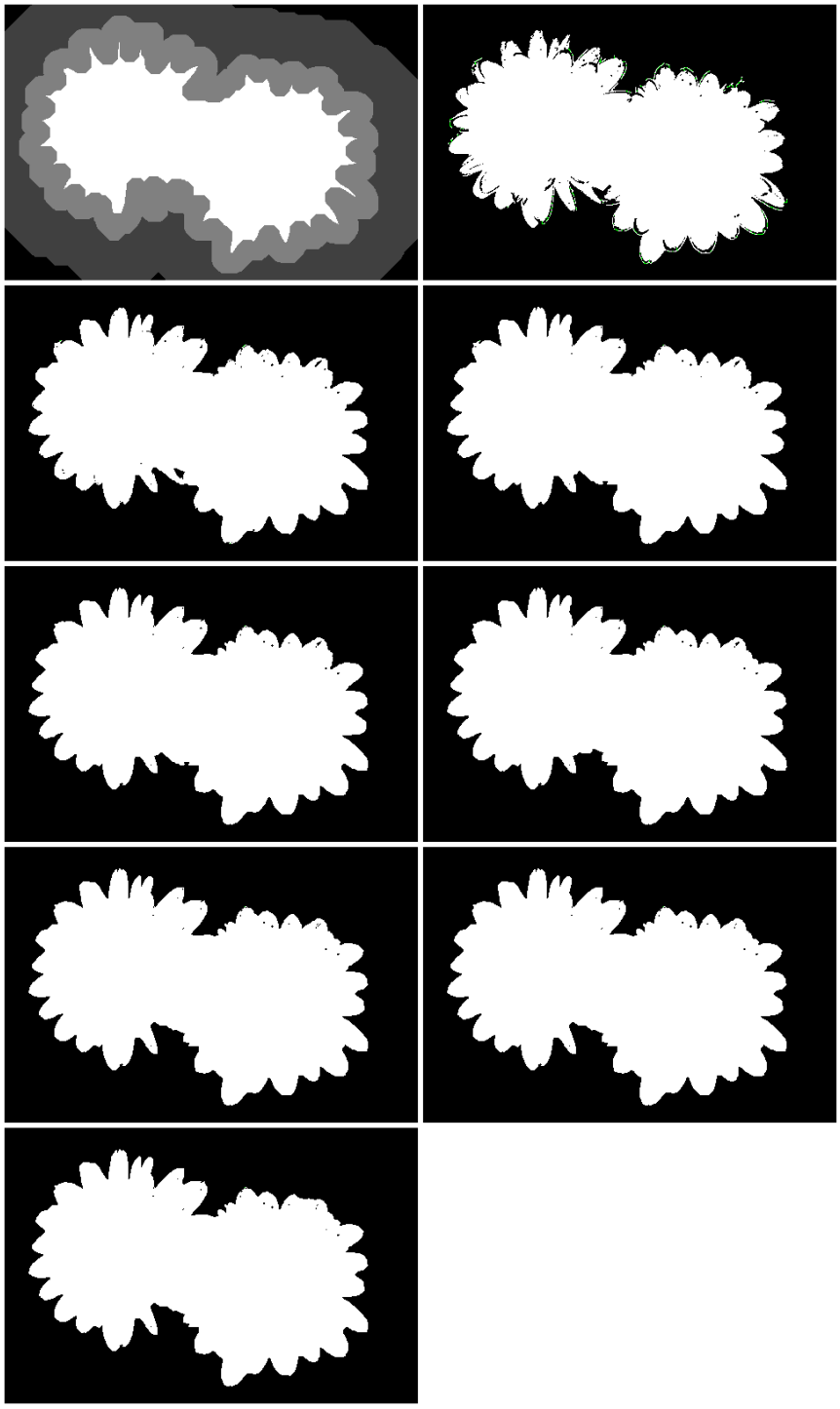


Image ”person2”, average overlap difference 0.04%

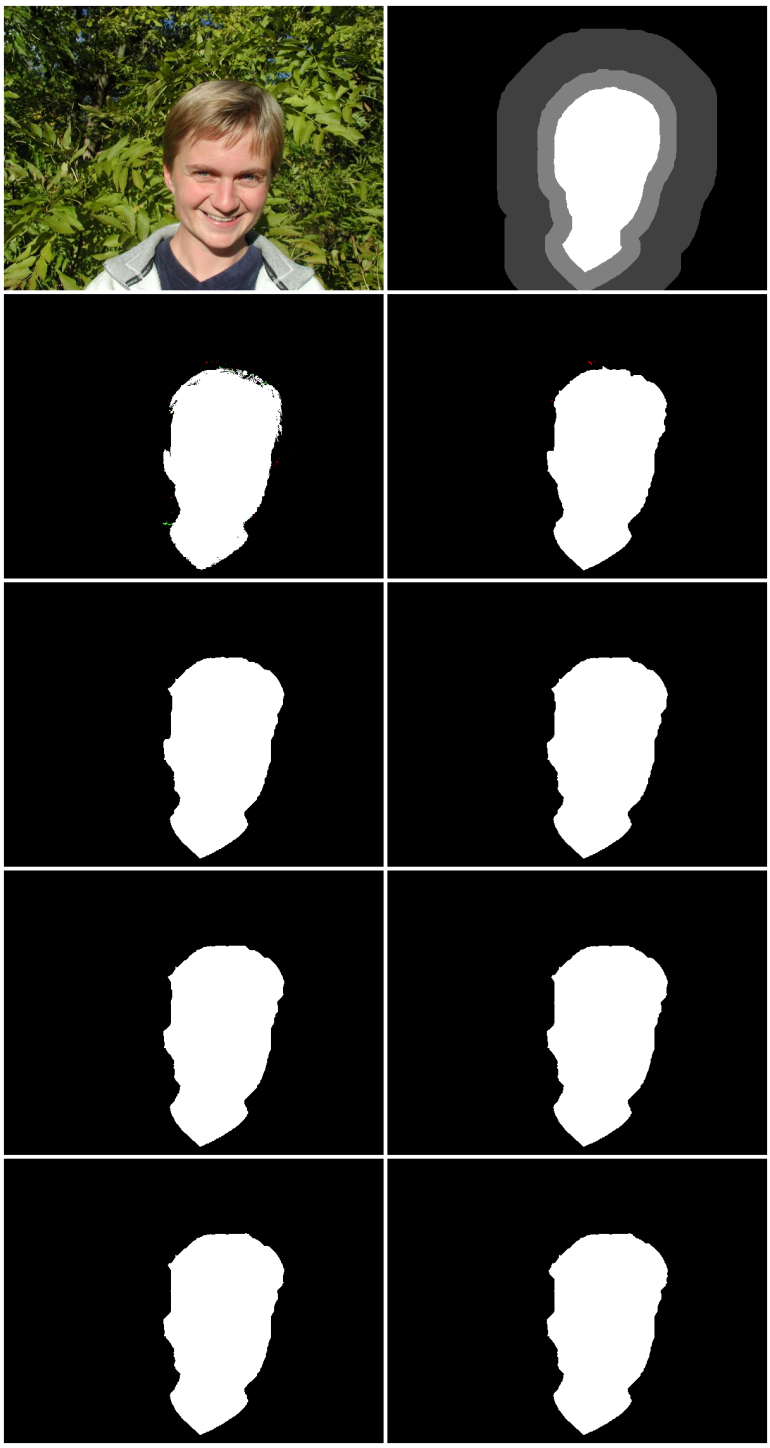


Image ”fullmoon”, average overlap difference 0.03%

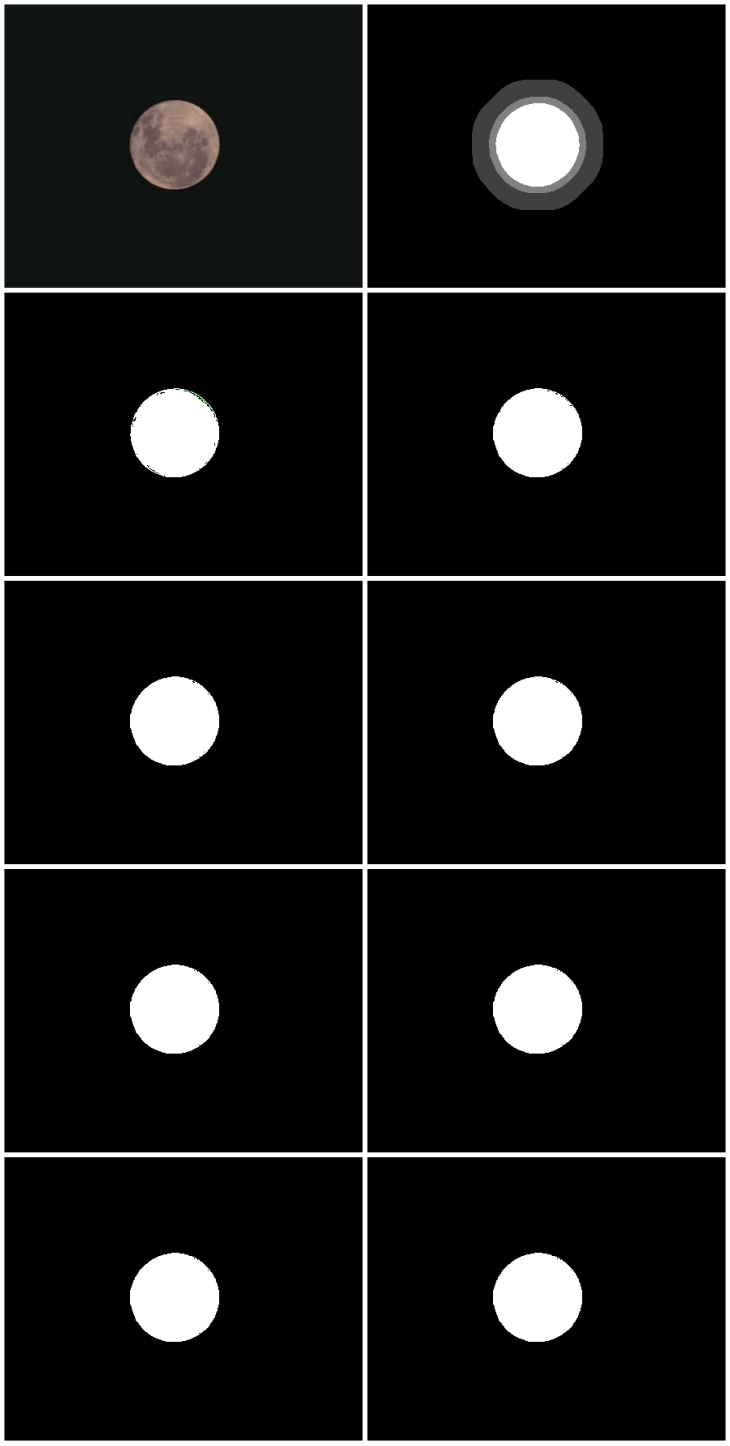


Image ”person3”, average overlap difference 0.03%

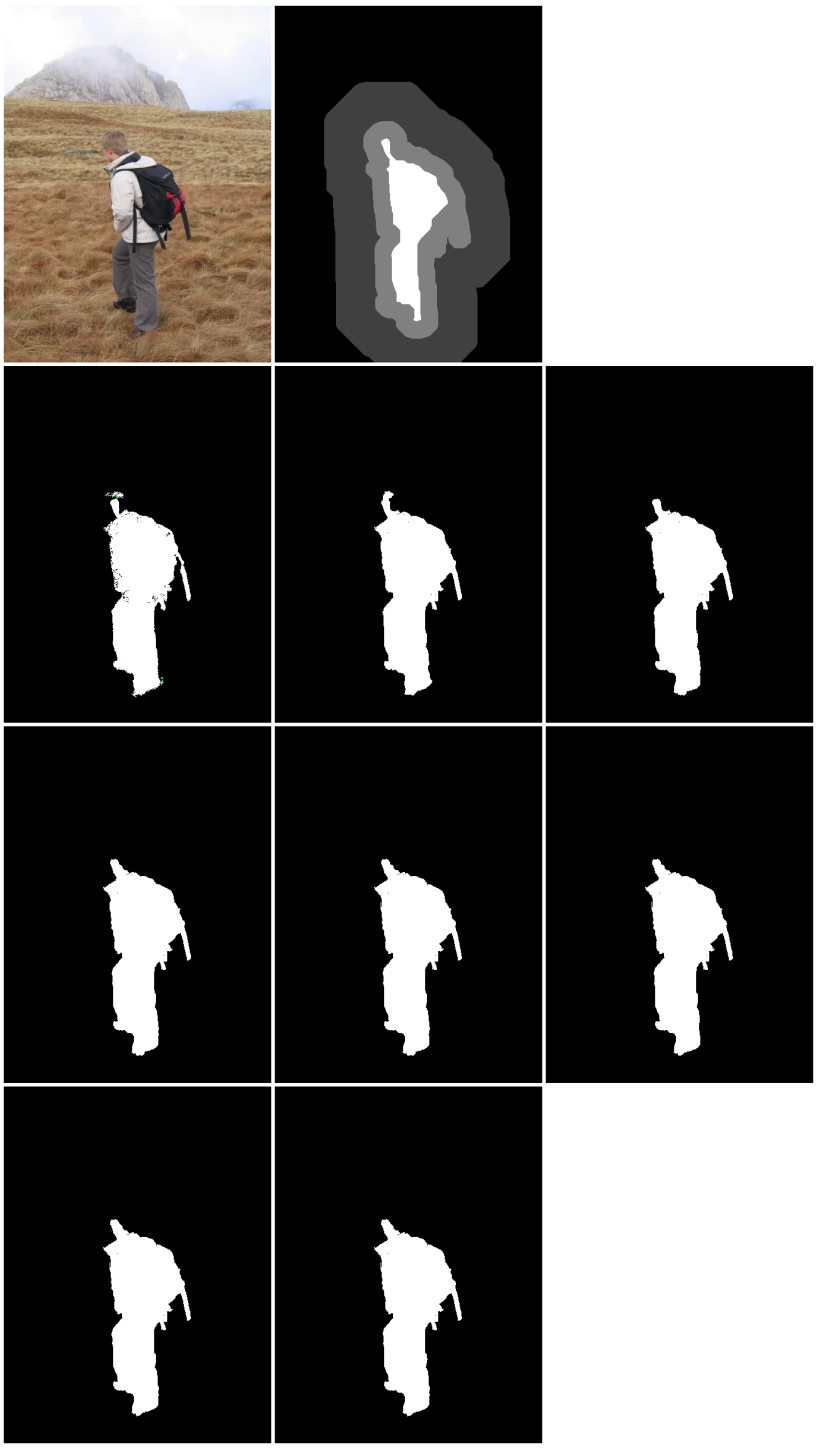


Image ”sheep”, average overlap difference 0.03%

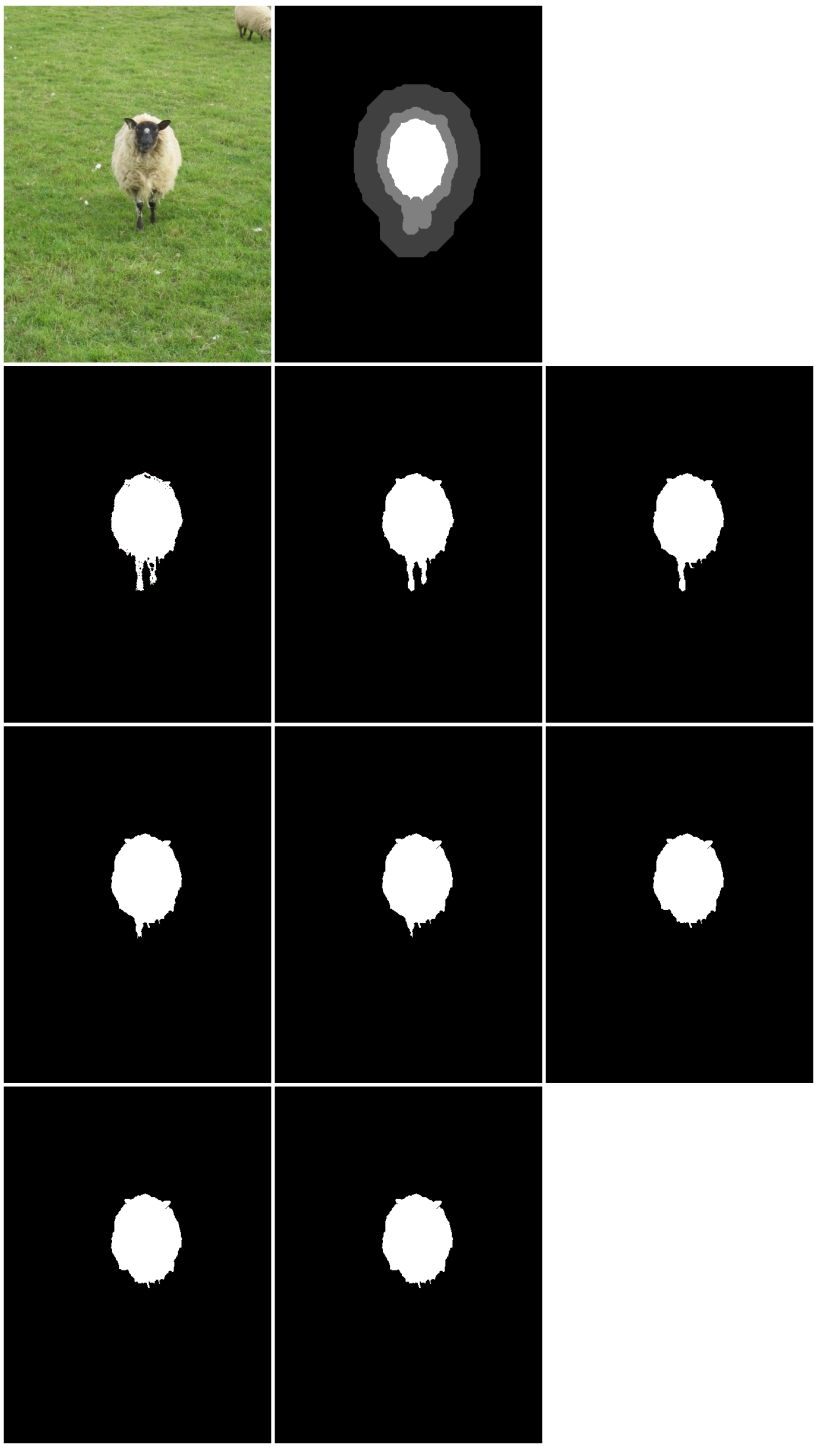


Image ”ceramic”, average overlap difference 0.02%

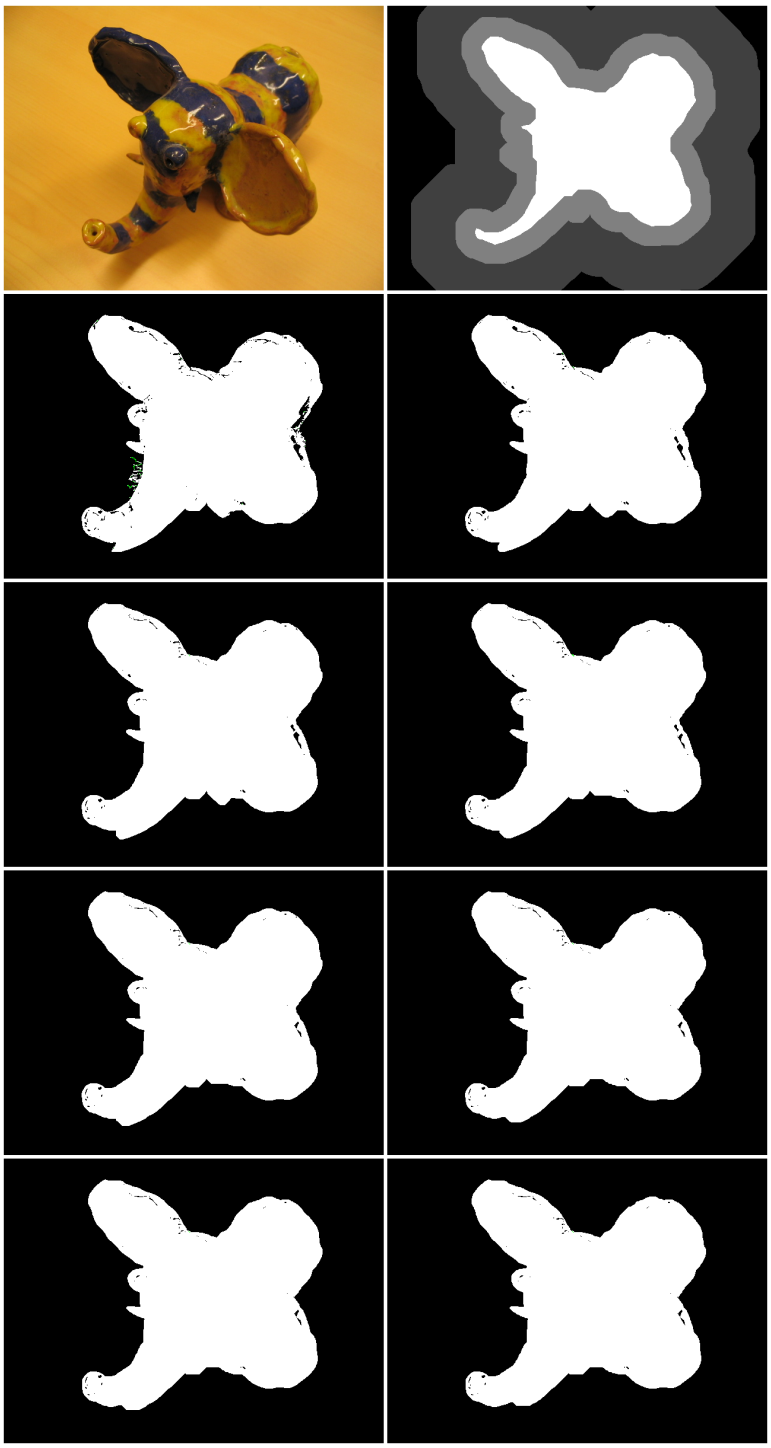


Image ”elefant”, average overlap difference 0.01%

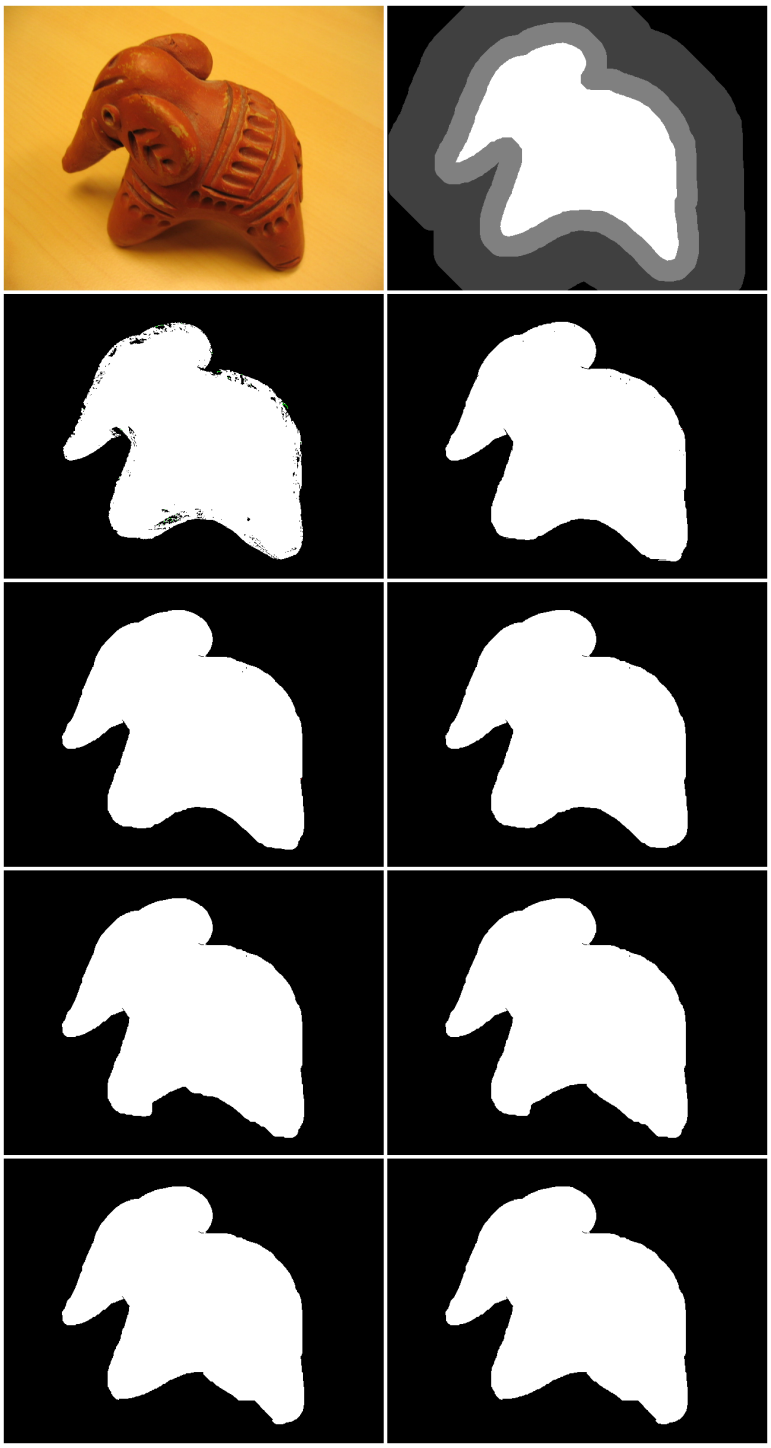


Image ”music”, average overlap difference 0.01%

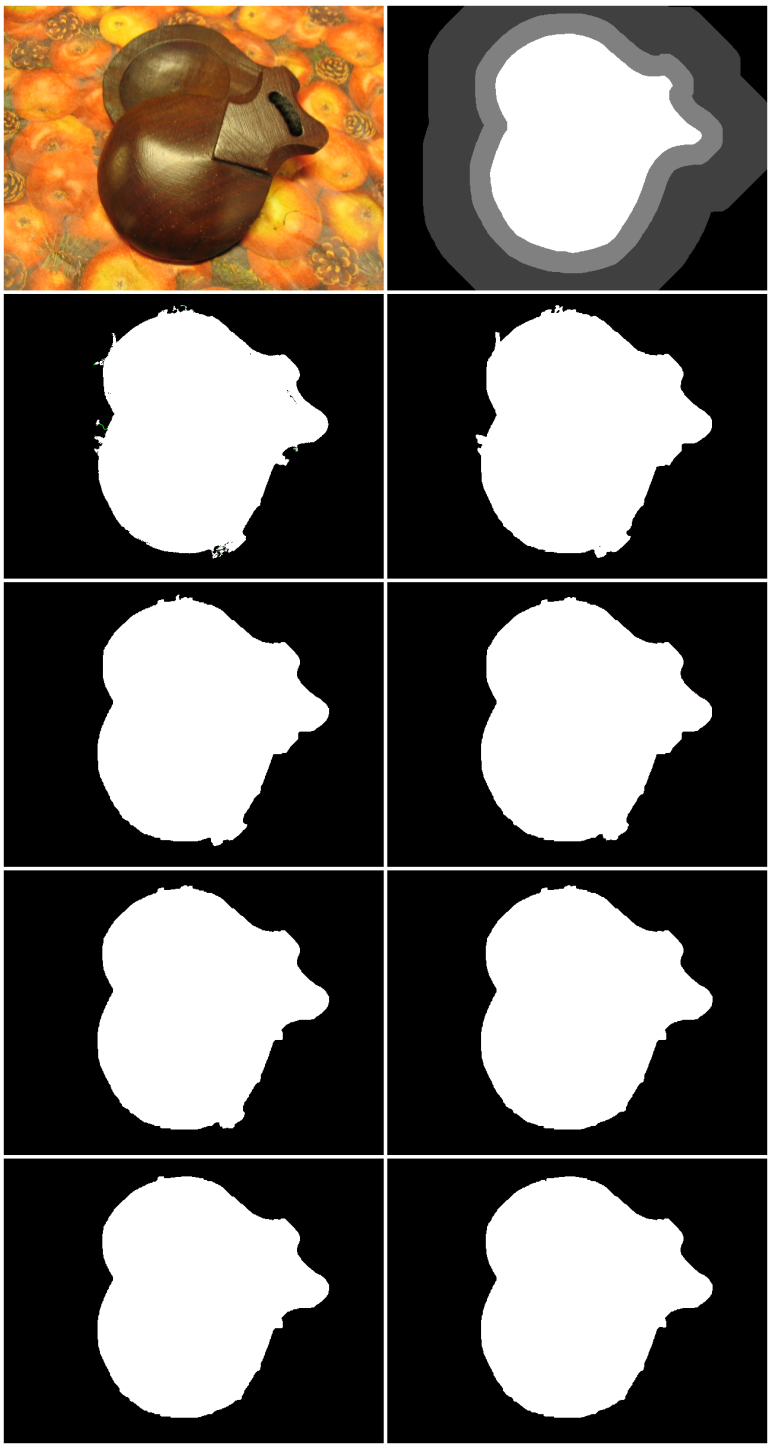


Image ”stone2”, average overlap difference 0.01%

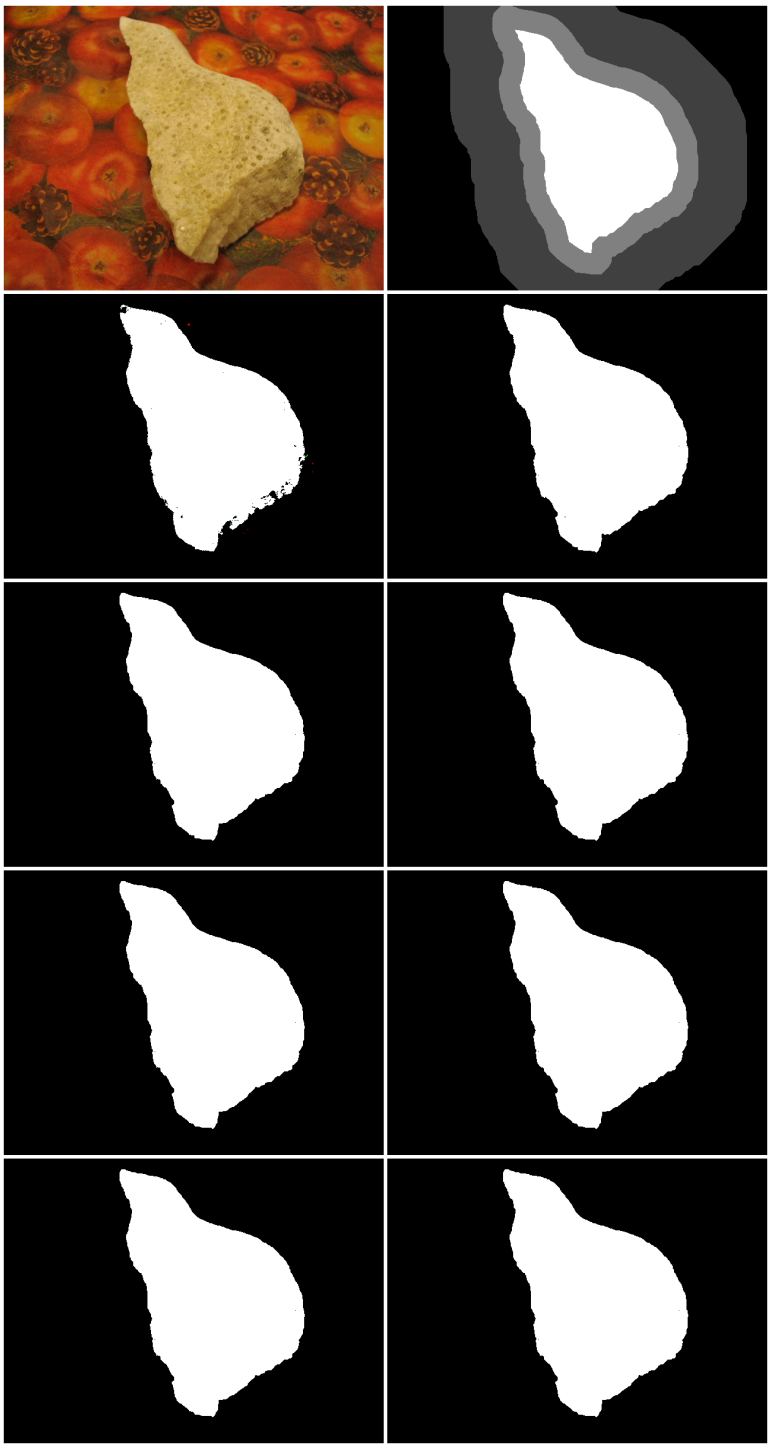


Image ”stone1”, average overlap difference <0.01%